Gamma-Ray Large Area Space Telescope (GLAST) Project

MOC Design Specification

Draft
Version 1.0

November 30, 2003

DOCUMENT APPROVAL

Prepared by:

____
Marilyn J. Mix, Omitron
MOC Software Engineer

Reviewed by:

____
Doug Spiegel, Omitron
MOC Manager

____
Dennis Small, GSFC Code 584
MOC Lead

____
Howard Dew, GSFC Code 581
Ground System Engineer

Approved by:

____
Mike Rackley, GSFC Code 581
Ground System/Operations Manager

Revision Status

This document is controlled by GLAST Ground System Configuration Control Board (CCB). Changes require prior approval of the MOC Manager. Proposed changes shall be submitted to the CCB via a Configuration Change Request (CCR).

VERSION	DATE	BY	DESCRIPTION
1.0	November, 2003	M. Mix	Original Draft Document

TABLE OF CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

1.1 Purpose and Scope

The purpose of this document is to describe the design of the GLAST Mission Operations Center (MOC), and will be used as a mechanism for design control through the GLAST Ground Segment documentation configuration management process.

The software, hardware, network, and facility design details are covered, including both internal and external interfaces. Some external interfaces are controlled by separate Interface Control Documents (ICDs), but all interfaces are described here. The MOC software design is defined to the process level. Detailed design of custom software will be captured separately in Software Development Folders for each component. A Data Dictionary, included as Appendix A, summarizes the external and internal data entities discussed in this document.

1.2 Applicable/Related Documents

The following documents were referenced during the development of this document. The reader is encouraged to use present and future versions of these documents for further research. Most of the documents are available on the GLAST Project web sites at http://glast.gsfc.nasa.gov/.

- *GLAST Ground System Requirements Document*, 433-RQMT-0006
- *GLAST Mission Operations Concept Document*, 433-OPS-001.
- *GLAST MOC Development Plan*, 492-MOC-001.
- *GLAST Spacecraft to MOC Interface Control Document*
- *GLAST MOC to GLAST Science Support Center (GSSC)/Instrument Operations Centers (IOCs) Interface Control Document (ICD*), 492-MOC-009.
- *GLAST MOC Functional & Performance Requirements Document*, 492-MOC-002.
- *GLAST MOC to Backup Ground Station Interface Control Document*, 492-MOC-010.
-
- *CCSDS 202.0-B-1 Blue Book*

1.4 Mission Overview

The GLAST mission involves the 5-year operation of the GLAST spacecraft and instruments to perform gamma-ray measurements over the entire celestial sphere with a sensitivity of a factor of 30 or more than obtained by earlier space missions. GLAST scientific objectives will be satisfied by two instruments. The main instrument, the Large Area Telescope (LAT), will have superior area, angular resolution, field of view, and dead time that together with the GLAST Burst Monitor (GBM) will provide a factor of 30 or more advance in sensitivity, as well as provide capability for study of transient phenomena. The GBM will have a field of view several times larger than the LAT and will provide spectral coverage of gamma-ray bursts that extends from the lower limit of the LAT down to 10 keV. With the LAT and GBM, GLAST will be a flexible observatory for investigating the great range of astrophysical phenomena best studied in high-energy gamma rays.

Spectrum Astro is responsible for the design and manufacture of the spacecraft, integration of the scientific instruments with the spacecraft, integration of the complete space vehicle/observatory with the Delta launch vehicle, and launch and early orbit activities. NASA/GSFC has program management responsibility. GLAST is an international collaboration of government agencies and academic institutions from the United States, France, Germany, Japan, and Sweden. The LAT is a joint project with NASA and the U.S. Department of Energy. The LAT will be constructed by Stanford University, the Stanford Linear Accelerator Center, the University of California, Santa Cruz, the Naval Research Laboratory, NASA Goddard Space Flight Center, and the international partners. The GBM is a joint project with Marshall Space Flight Center, the University of Alabama, and the Max-Planck Institute in Germany. Universal Space Network of Horsham, PA is scheduled to support GLAST using their South Point, Hawaii and Dongara, Australia ground stations as a backup sites for spacecraft telemetry downlink and command uplink.

The GLAST spacecraft will be launched on a Boeing Delta II 2920H-10 vehicle from the Eastern Test Range (ETR), Florida. A launch date of February 2007 is currently scheduled. GLAST will then settle into an orbit at an altitude of 565 km and an inclination of 28.5 degrees, orbiting the Earth once every 96 minutes. Following a 60-day checkout period, the nominal 5-year science mission will commence.

1.5 Ground Segment Architecture Overview

The GLAST Ground System provides for:

- Radio Frequency (RF) communications with the spacecraft
- Spacecraft and instrument monitoring and control
- GRB alert notification
- Mission planning and scheduling
- Science data processing
- Science data archive and distribution

These functions are performed by existing and new facilities. A GLAST Ground System mission architecture overview is shown in Figure 1.5.1.

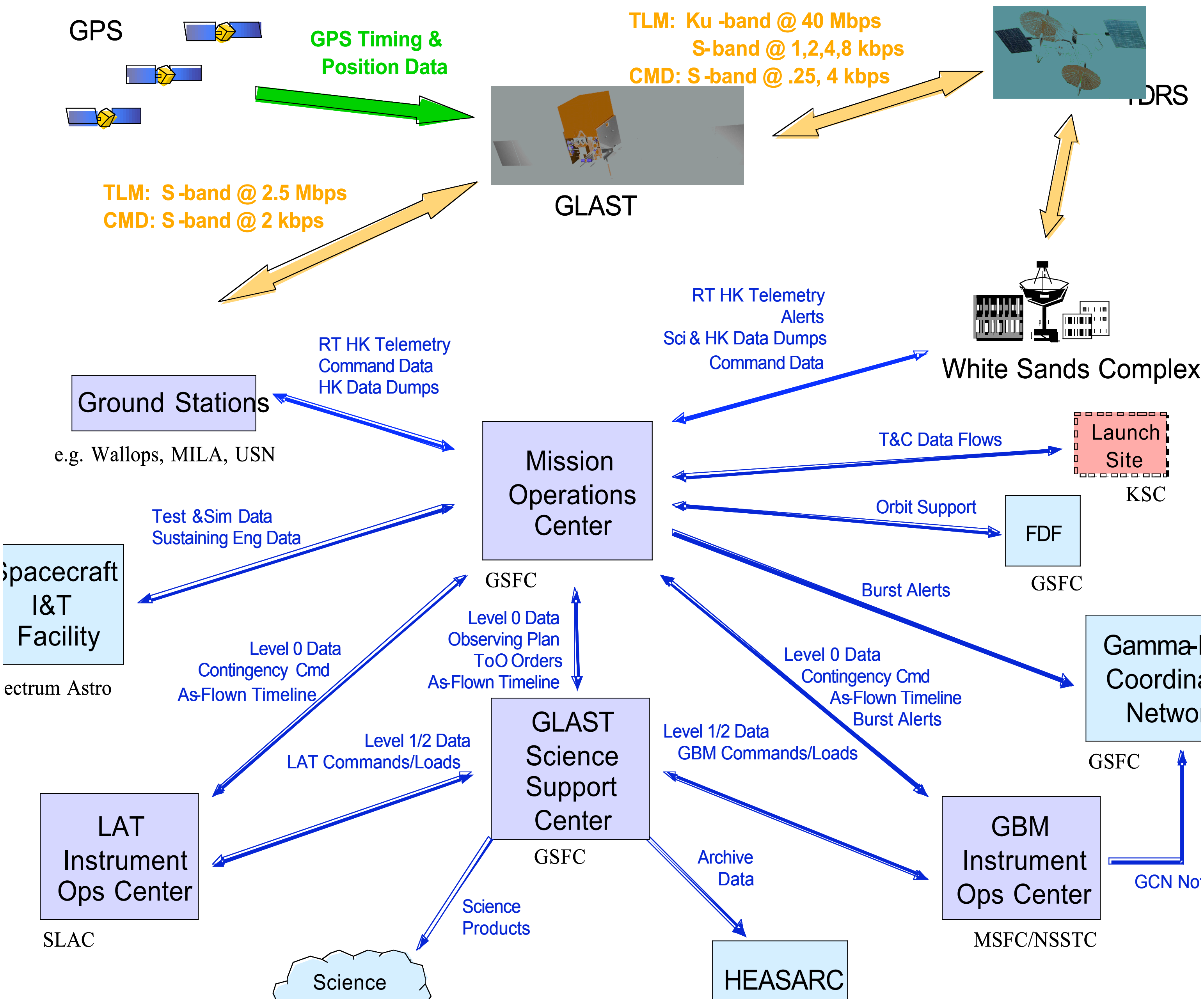


Figure 1.5.1 - GLAST Ground Segment Architecture Overview

1.5.1 RF Communications

The primary RF communications interface for GLAST is the Space Network (SN) Tracking and Data Relay Satellite System (TDRSS). TDRSS provides Ku-band and S-band forward and return link services for command and telemetry transmission between the GLAST observatory and the MOC, via the White Sands Complex (WSC) located in Las Cruzes, NM. Backup RF communications support is provided by the Ground Network (GN), including the ground station located in Wallops Island, VA, and Merritt Island (MILA), FL, and the Universal Space Network (USN) commercial ground stations located at South Point, Hawaii, and Dongerra, Australia. The GN provides S-band forward and return link services only.

1.5.2 Ground Communications Network

The ground communications network is provided by the NASA Integrated Services Network (NISN), which provides data and voice connectivity between the MOC and the WSC, the Flight Dynamics Facility (FDF), Wallops Ground Station, and the GCN. Communication between the MOC and the GSSC, the LIOC, the GIOC, and the USN Network Management Center (NMC) are accomplished over the Internet.

1.5.3 GLAST Mission Operations Center (MOC)

The MOC, located at Goddard Space Flight Center, will operate the GLAST satellite and instruments. The MOC will support pre-launch operations, launch and 60-day checkout, and normal and contingency operations throughout the life of the mission. Spectrum Astro provides spacecraft bus training for the Flight Operations Team (FOT) pre-launch, and is responsible for spacecraft launch processing and initial 60-day post-launch checkout.

The MOC performs all spacecraft and instrument mission planning, commanding, monitoring, and Level 0 data processing and delivery to the GSSC, LIOC, and GIOC. The MOC provides rapid response for the follow-up of new GRBs through Target of Opportunity (ToO) orders from the GSSC. The MOC incorporates automation of spacecraft operations and data processing to permit a small operations team and "lights-out" operation, and to speed data processing and response to GRBs and ToOs. An archive for the life of the mission is provided of all raw telemetry data, command transmissions, and ancillary data.

1.5.4 GLAST Science Support Center (GSSC)

The GLAST Science Support Center (GSSC) is located in GSFC's Laboratory for High Energy Astrophysics. The GSSC supports over-all science planning, analysis software distribution, and data distribution. It is the primary conduit for the weekly science activity plan and for any ToO orders or instrument loads to be sent to the MOC. The GSSC develops the science activity plan and includes inputs from the GIOC and LIOC. The GSSC, GIOC, and LIOC can all access the GLAST Level 0 data produced by the MOC.

The GSSC assists the science community in the scientific analysis of GLAST data. The GSSC has the lead role in developing the software tools needed to convert GLAST telemetry into FITS files and to perform scientific analysis of the GLAST data. After launch, the GSSC updates the analysis tools as the understanding of the techniques utilized improves with experience. In addition, the GSSC maintains documentation of the GLAST data and results, provides user guides, provides online data analysis recipes, and provides other information for use by the science community.

1.5.5 GRB Coordinates Network (GCN)

The GCN distributes location and light curve information for GRBs detected by all spacecraft capable of detecting GRBs to interested members of the science community. The GCN receives GRB alerts from Burst Alert Processor (BAP) system running in the MOC. The GCN provides rapid dissemination of GLAST alerts and finder fields to enable ground observatories and operators of other spacecraft to plan correlative observations. The instrument teams will subscribe to the GCN to receive notification of GRBs detected by other spacecraft.

1.5.6 HEASARC/Data Archival

The High Energy Astrophysics Science Archive Research Center (HEASARC) is NASA's archive for the high energy astrophysics missions in which NASA had a role. The GSSC's computer system is part of the HEASARC system, and the GSSC's databases will reside on the HEASARC's storage media. The software tools that will be provided to the scientific community for the analysis of GLAST data are being developed in HEADas, the HEASARC's software system, which the HEASARC is committed to maintain. The HEASARC will become the permanent archive of GLAST's data, response functions, analysis software and documentation. GLAST's association with the HEASARC will guarantee the long-term accessibility of the GLAST data, and will ensure a multi-wavelength approach to the analysis of these data.

1.5.7 Spacecraft Sustaining Engineering Facility

Spectrum Astro, in Gilbert, AZ, will provide sustaining engineering support of the spacecraft bus, including flight software maintenance, trending and analysis of spacecraft subsystems, and anomaly resolution support. Spectrum systems and subsystems engineers will access spacecraft SOH data from the MOC, and provide flight software loads to the MOC for uplink to the spacecraft.

1.5.8 LAT Instrument Operations Center (LIOC)

The LIOC is located at the Stanford Linear Accelerator Center (SLAC) in Palo Alto, CA. The responsibilities of the LIOC include data verification, LAT health and safety monitoring, LAT command generation, flight software validation and maintenance, and alert processing and logging.

1.5.9 GBM Instrument Operations Center (GIOC)

The GIOC is located at the University of Alabama-Huntsville, AL. The responsibilities of the GIOC include data verification, GBM health and safety monitoring, GBM command generation, flight software validation and maintenance, and alert processing and logging. The GIOC is a collaborative effort between the National Space Science and Technology Center in the U.S. and the Max Planck Institute for Extraterrestrial Physics (MPE) in Germany.

2.0 MOC SOFTWARE DESIGN

2.1 MOC Functional Overview

The GLAST Mission Operations Center (MOC), located at GSFC, is the central hub for mission operations. The MOC performs all spacecraft and instrument monitoring and control, mission planning and scheduling including spacecraft and instrument data capture, archive and distribution. In addition, the MOC coordinates and schedules all external support for mission operations. The MOC also includes remote access capability to review spacecraft, instrument, and ground system data and status. The remote engineering stations will be used for on-call Flight Operations Team (FOT) support and sustaining engineering support from the spacecraft and instrument team facilities. The MOC architecture is composed of commercial- and government-off-the-shelf (COTS/GOTS) hardware and software that is tailored for GLAST mission support. Custom software is provided where the functionality cannot be provided by existing packages.

The functional data flow model is described in the following sections using Context Diagrams, external interface descriptions, data flow diagrams and process descriptions. The Data Dictionary, which defines the terms used in the figures, is located in Appendix A.

2.1.1 MOC External Interfaces

Figure 2.1.1 depicts the MOC external interfaces in a context diagram (Level-0 DFD).

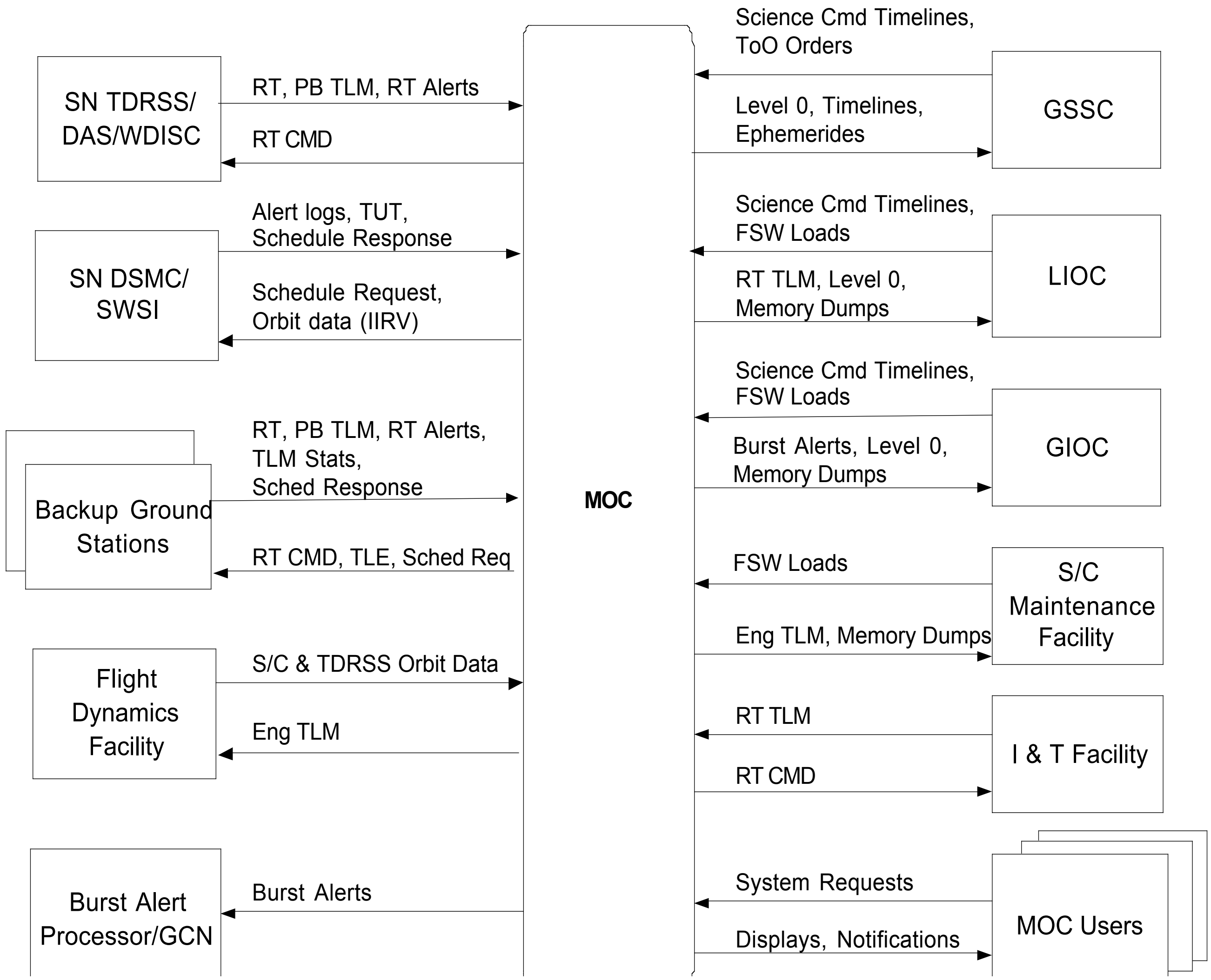


Figure 2.1.1 - MOC Context Diagram

The MOC external interfaces are described below:

- SN TDRSS/DAS/WDISC – The Space Network (SN) provides communication services through the TDRSS. GLAST is using the DAS system for real-time return link low-rate data for burst alert and emergency messages, and spacecraft housekeeping data. Commands to GLAST through the TDRSS link are accomplished via the White Sands Complex (WSC) Transmission Control Protocol TCP/IP Data Interface Service Capability (WDISC). High-rate data is received by the GLAST-provided Ku-band front-end at WSC.

- SN DSMC/SWSI – Scheduling of SN services, including DOWD tracking data, and obtaining performance data is accomplished through the SN Web Services Interface (SWSI), which provides the interface to the Data Systems Management Center (DSMC) at GSFC.

- Backup Ground Stations – The Wallops Island Ground Station, Merritt Island Ground Station (MILA), and Universal Space Network (USN) provide backup RF communication support for telemetry and command. The USN backup commercial ground stations are located in Hawaii and Australia.

- Flight Dynamics Facility – The Flight Dynamics Facility (FDF) provides orbit determination services for GLAST during the early mission phase. For the remainder of the mission, the spacecraft position is derived from the GPS-based position data in telemetry.

- Burst Alert Processor/Gamma-Ray Coordinates Network – The MOC provides burst alert messages to the GCN via the Burst Alert Processor (BAP) provided by the GIOC.

- GSSC – The GLAST Science Support Center provides science-oriented commands and loads to the MOC. The MOC provides products to the GSSC for planning and for processing the science data into higher level science products that are distributed to the science community.

- LAT Instrument Operation Center – The LAT science instrument team provides FSW maintenance and sustaining engineering services during mission operations. FSW loads are provided to the MOC for uplink to the spacecraft. Science commands are normally sent to GSSC for coordination, however they can be sent directly to the MOC as a contingency. The MOC provides realtime telemetry and Level 0 data sets to the LIOC.

- GBM Instrument Operation Center – The GBM science instrument team provides science commands, FSW maintenance, and sustaining engineering services during mission operations. FSW loads are provided to the MOC for uplink to the spacecraft. Science commands are normally sent to GSSC for coordination, however they can be sent directly to the MOC as a contingency. The MOC provides Level 0 data sets and burst alerts to the GIOC.

- S/C Maintenance Facility – The spacecraft manufacturer, Spectrum Astro in Gilbert, AZ, provides Flight Software (FSW) maintenance and sustaining engineering services during the mission operations. FSW loads are provided to the MOC for uplink to the spacecraft.

- I&T Facility – Pre-launch, Spectrum Astro provides the Integration and Test Facility located in Gilbert. AZ. During testing, the MOC sends realtime commands to and receives realtime telemetry from this facility.

- MOC Users – The users of the MOC include the Flight Operations Team (FOT), the S/C Vendor, and the Science Instrument Teams. The users can view real-time telemetry data and events, and access stored telemetry data. Authorized users can perform mission scheduling activities and commanding.

The MOC Context Diagram data flows are described in the Table 2.1.1 – MOC Context Diagram Data Flow Definitions. The document references for the external interfaces of the GLAST MOC are listed in Table 2.1.2 – GLAST MOC External Interfaces Document References.

Table 2.1.1 – MOC Context Diagram Data Flow Definitions

Data Item	Source	Destination	Description
RT Tlm	Backup Ground Stations	MOC	Realtime telemetry (VC 0)
PB Tlm	Backup Ground Stations	MOC	Playback telemetry files via ftp
TLM Stats	Backup Ground Stations	MOC	Ground Station Performance Data (Network Status)
RT Alerts	Backup Ground Stations	MOC	GRB & S/C emergency alerts
Schedule Response	Backup Ground Stations	MOC	Ground station contact schedule
RT CMD	MOC	TDRSS/WDISC, Backup Ground Stations	Realtime commands, Stored Command Loads, Memory Loads
TLE	MOC	Backup Ground Stations	NORAD two-line elements
Schedule Request	MOC	Backup Ground Stations	Ground station contact or retransmission request
RT Tlm	SN TDRSS/DAS/ WDISC	MOC	Realtime telemetry (VC 0)
PB Tlm	SN TDRSS/DAS/ WDISC	MOC	Playback telemetry files
RT Alerts	SN TDRSS/DAS/ WDISC	MOC	GRB & S/C emergency alerts
Schedule Request	MOC	SN DSMC/SWSI	TDRSS Fwd link schedule requests, via SWSI interface
Orbit data (IIRV)	MOC	SN DSMC/SWSI	Spacecraft orbit data
Alert logs	SN DSMC/SWSI	MOC	SWSI alerts
TUT	SN DSMC/SWSI	MOC	TDRSS Unscheduled Time Report
Schedule Response	SN DSMC/SWSI	MOC	TDRSS schedule data
Displays, Notifications	MOC	MOC User	Responses to system requests, anomaly notification, data/displays, e-mail, e-page
System Requests	MOC User	MOC	Authorized MOC user requests
ToO Order	GSSC	MOC	Approved Target of Opportunity

Data Item	Source	Destination	Description
Science Cmd Timeline	GSSC, GIOC, LIOC	MOC	Time-tagged list of science commands to be included in a stored command load
Level-0	MOC	GSSC, LIOC, GIOC	Level-0 processed telemetry
Timelines	MOC	GSSC	Projected mission events timeline, As-flown timeline
Ephemerides	MOC	GSSC	Spacecraft and TDRSS ephemeris files
S/C & TDRSS Orbit Data	Flight Dynamics	MOC	Spacecraft and TDRSS orbit data
FSW Load	S/C Maintenance Facility, GIOC, LIOC	MOC	Flight software loads requested for uplink
Engineering TLM	MOC	FDF, S/C Maintenance Facility	Housekeeping sequential prints
Memory Dumps	MOC	S/C Maint. Facility, LIOC, GIOC	S/C OBC and Instrument processor memory dumps
Burst Alerts	MOC	BAP, GIOC	GRB Alert Packets

Table 2.1.2 – GLAST MOC External Interfaces Document References

Interface	Document	Ref ID
MOC – GSSC/GIOC/LIOC/BAP	GLAST Operations Data Products ICD	492-MOC-0xx
MOC - TDRSS DAS	ICD Between the DAS and the DAS Customers	453-ICD-DAS/Customer
MOC - TDRSS WDISC	Interface Control Document Between the Network Control Center Data System and the Mission Operations Centers	451-ICD-NCCDS/MOC
	WSC Transmission Control Protocol (TCP)/Internet Protocol (IP) Data Interface Service Capability (WDISC) User's Guide	452-WDISC-UG 98
MOC - SWSI	SWSI User's Guide	
MOC - Flight Dynamics Facility	Interface Control Document Between the FDF and the GLAST MOC	
MOC - Spacecraft Vendor Facility	GLAST Spacecraft to Mission Operations Center (MOC) [DS2]Interface Control Document	
MOC to Backup Ground Station	GLAST MOC to Backup Ground Station Interface Control Document	492-MOC-010
MOC to MOC Users	MOC User Guide	

2.1.2 MOC Software Functional Architecture

The MOC supports all activities necessary to plan mission operations, operate the spacecraft and instruments, and evaluate their performance and state over the mission lifetime. Mission operations will consist of a well-defined set of procedures using the various software packages to generate the necessary products at each point in the planning process. The final products of the planning process, scripts containing spacecraft, instrument, and ground segment control commands, will be used to provide fully automated real time contacts with the spacecraft.

The following software functional architecture diagram identifies five major functions performed in the MOC. The internal and external interfaces for these activities are illustrated in Figure 2.1.2 – MOC Software Functional Architecture and a description of these data entities are defined in the Data Dictionary in Appendix A. Descriptions of the internal interface definitions are in Appendix B.

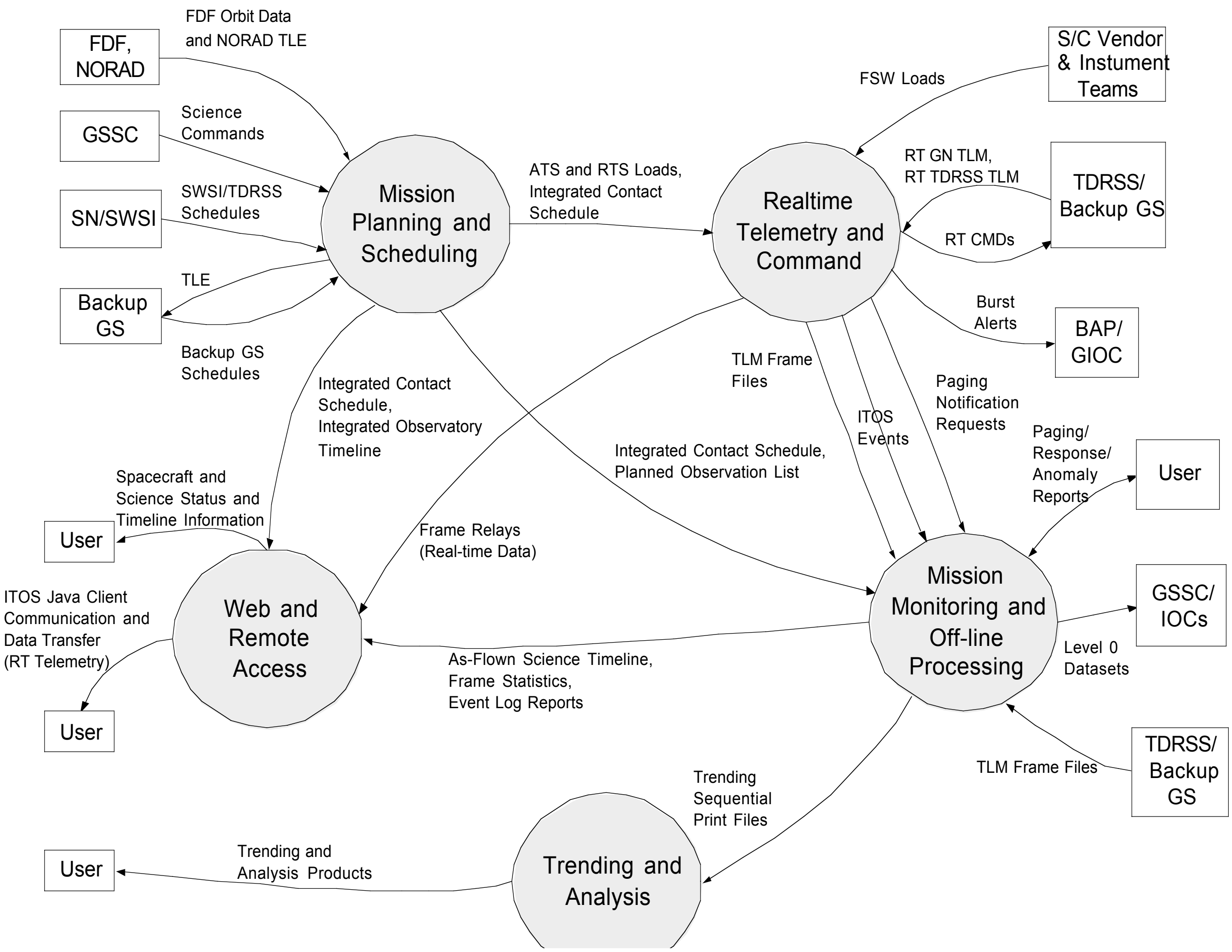


Figure 2.1.2 - MOC Software Functional Architecture

The five functional areas comprising the MOC are the following:

- Mission Planning and Scheduling
  The Mission Planning and Scheduling subsystem produces a planned set of spacecraft commands needed to perform science observations and spacecraft housekeeping. This subsystem creates a science timeline by combining science commands from the GSSC with predicted orbital events and contact schedules. To support science planning, the subsystem manages contact scheduling and merges contact schedules from various sources into a single contact schedule. It also produces the orbital event products based on GPS data in the telemetry. Finally, the science commands and additional spacecraft engineering commands are merged into time-tagged stored command loads. These loads are ready to be uplinked to the spacecraft where the commands will be executed.
  A more detailed description of this subsystem is presented in Section 2.4.

- Real-time Telemetry and Command
  This subsystem is responsible for the receipt, processing, distribution and storage of real-time telemetry received from the spacecraft via primary TDRSS and as a backup the ground stations. This subsystem is also responsible for supporting the command activities including management of command loads, transmission of commands, and verification of commands.
  A more detailed description of this subsystem is presented in Section 2.2.

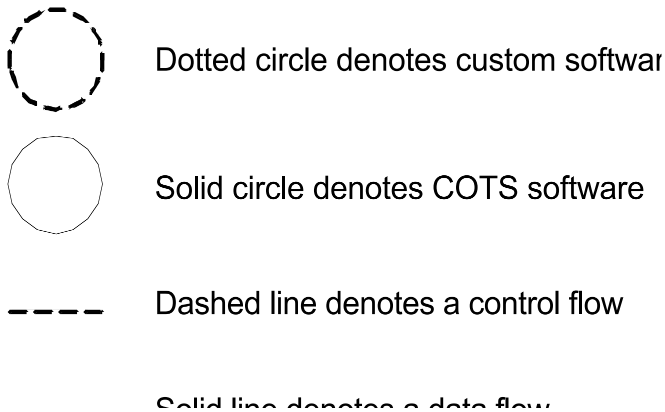
- Mission Monitoring & Offline Processing
  This subsystem's main responsibility is the processing of playback telemetry, collecting and logging telemetry statistics and the archiving of telemetry and MOC log files. Playback telemetry files for virtual channels 0, 1, 2, 3, 8, 9, 10, and 11 are processed to generate decommutated and Level 0 products. Data products are distributed to the GSSC and IOCs. While generating these products it also determines missing frames and other frame accountability statistics, logging this information to the MOC file server.
  A more detailed description of this subsystem is presented in Section 2.3.

- Trending & Analysis
  This subsystem performs trending and analysis of downlinked telemetry data as needed for the evaluation of spacecraft and instrument health and safety.
  A more detailed description of this subsystem is presented in Section 2.5.

- Web and Remote Access
  The Web and Remote Access subsystem is the primary portal for outgoing data and information from the MOC to other ground segment elements and to external public users. The subsystem provides capability for external users to view real-time mission telemetry data via an ITOS JAVA client. It also provides interfaces to view mission documentation, reports, data, and statistics.
  A more detailed description of this subsystem is presented in Section 2.6.

2.1.3 Figure Conventions

The data flow and process diagrams in the following sections employ this convention:



2.2 Real-time Telemetry and Command

The Real-time Telemetry and Command subsystem consists of three software components: the Integrated Test and Operations System (ITOS), Automation Monitoring and Control software, and the Event Delogger. These components are depicted below in Figure 2.2 MOC Real-time Telemetry and Command Software Flow Diagram, along with their external and internal interfaces.

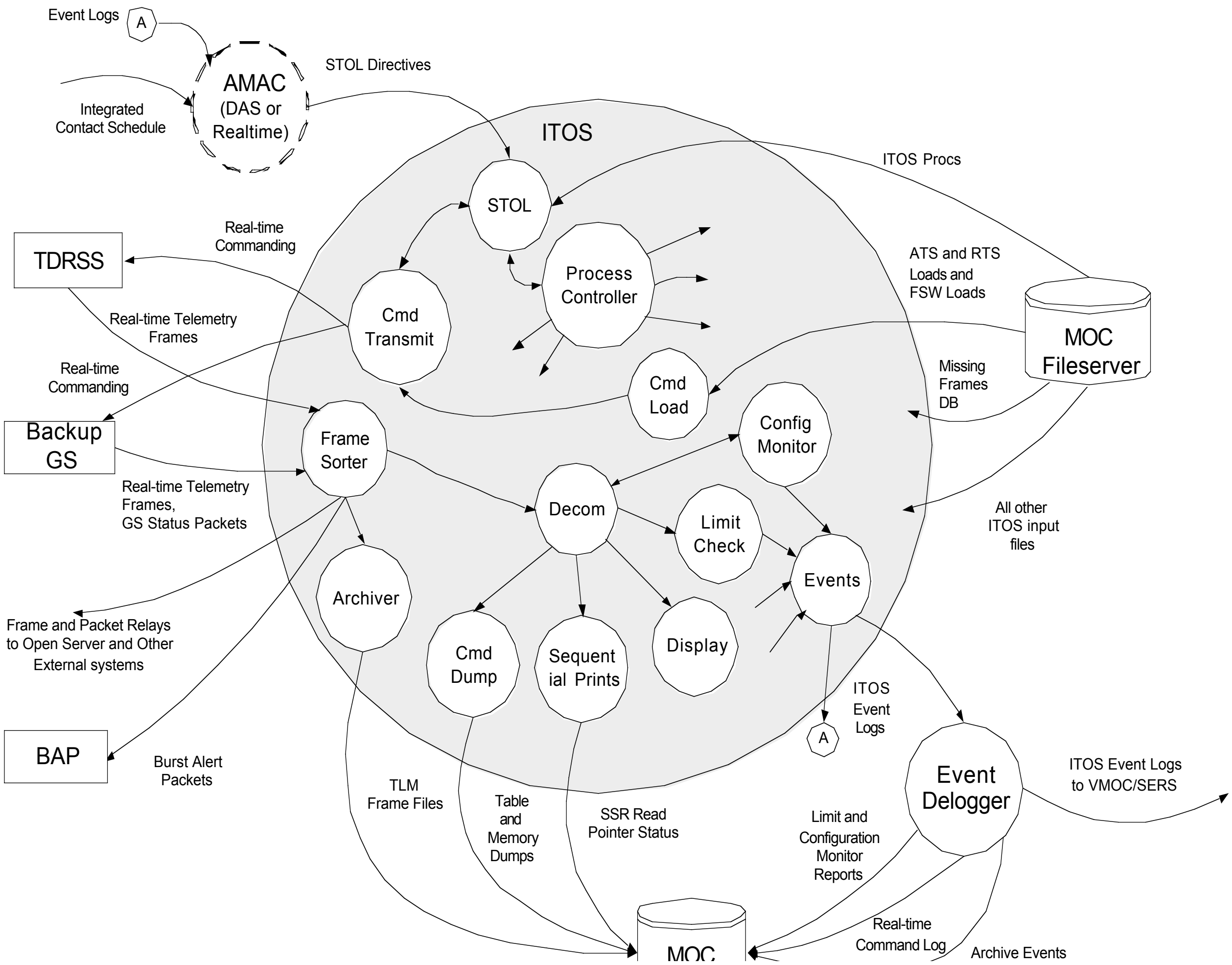


Figure 2.2 - MOC Real-time Telemetry and Command Software Flow Diagram

2.2.1 Integrated Test & Operations System (ITOS)

The Integrated Test and Operations System (ITOS) is a Government Off The Shelf (GOTS) suite of software developed by a group in the Real-time Software Engineering Branch at the Goddard Space Flight Center for control of spacecraft and spacecraft components during development, test, and on-orbit operations. It is a low-cost, portable, highly configurable system, which runs under a variety of UNIX operating systems, including Solaris, FreeBSD, and Linux on workstation or PC hardware.

The ITOS design consists of a cluster of workstations interconnected over a local area network. Each workstation runs the complete ITOS software, with one designated as the primary operator console. This console receives data from and sends commands to and from the spacecraft interface over an IP Ethernet connection. The primary console feeds the data it receives from the spacecraft interface to all other ITOS workstations. Each ITOS workstation unpacks the data packets and performs data processing tasks such as limit checking, engineering unit conversions, and configuration monitoring. ITOS has an event module, which recognizes spacecraft events, logs them, and forwards them to operators or external programs for processing. In addition, the primary ITOS console can distribute data via an IP Ethernet to other systems or software attached to the network.

Key features of the ITOS software for GLAST are:

- Distributed software architecture -- runs on one machine or a cluster of machines.
- Database driven – highly configurable without software modifications.
- Automated operations via STOL procs.
- State modeling via configuration monitors.
- Archive and playback CCSDS frame or packet telemetry data.
- Display data via traditional display pages.
- Display data as strip charts or plots.
- Display raw transfer frames.
- Display raw packets.
- Display data via Java applets over the World Wide Web.
- Capture data via sequential prints.
- Output data as CCSDS transfer frames to external applications.
- Output data as CCSDS packets to external applications.
- Output data as mnemonic values to external applications.
- Maps non-CCSDS data such as station status packets to CCSDS transfer frames and packets.
- Ingests telemetry using TCP/IP or UDP/IP network protocols.
- Accepts telemetry as packets or transfer frames in a variety of encapsulations.
- Telemetry data processed in software, not custom hardware.
- Supports CCSDS COP-1 commanding.
- Outputs commands using TCP/IP or UDP/IP network protocols.
- Outputs commands in a variety of encapsulations.
- Unix based. Known to build and run correctly on: Sun Solaris, Sun Solaris X86, and Linux on i386
- HTML-based help and telemetry and command database viewing.

The major components of ITOS consist of the database module, the command module, the telemetry module, the display module, and STOL, the interface that runs the ITOS software.

Database module

The database describes spacecraft commands and spacecraft telemetry. Each spacecraft command is described in terms of the command mnemonic, zero or more command fields, and submnemonic values. Spacecraft telemetry is described in terms of packet maps and mnemonics: a mnemonic is an individual data point, the packet maps describes where and from which packet to extract the mnemonic. Telemetry mnemonics may have associated red and yellow high/low limits and analog or discrete conversions. ITOS also includes tools for viewing the telemetry and command database over the World Wide Web.

The database module includes the current value table (CVT), which contains the most recent value for each mnemonic. Most ITOS components obtain mnemonic values by polling the current value table. An alternative mechanism, where the component registers interesting mnemonics and gets notified every time a new value arrives, is also available.

Command module

The command module receives spacecraft command inputs from STOL. The spacecraft commands include spacecraft and instrument memory and table loads and real-time spacecraft commands. Status and operation of the command module is reported in display pages and detailed in the ITOS event log.

The command processing system supports the Consultative Committee for Space Data Systems (CCSDS) Command Operation Procedure (COP), COP-1 Enhanced Service. COP-1 fully specifies the closed-loop protocol executed between the sending (ground) and receiving (spacecraft) entities. The COP consists of a pair of synchronized procedures: a Frame Operation Procedure (FOP) that executes within the sending entity; and a Frame Acceptance and Reporting Mechanism (FARM) that executes within the receiving entity. The FOP transmits telecommand transfer frames to the FARM. The FARM returns telemetered Command Link Control Words (CLCW's) within the telemetry transfer frame trailers to the FOP, which provides return status about the acceptance of the frames by the spacecraft.

COP-1 operates on the principle of sequential frame acceptance and retransmission, with frame sequence numbering. The FOP initiates the transmission of TC Frames whose sequence numbers are arranged in ascending sequential order. The FARM only accepts frames if their sequence numbers match the expected ascending order. As soon as a sequence error is encountered, the FARM rejects all subsequent frames whose sequence numbers do not match the expected order. The FOP monitors the CLCW to determine if frames are being rejected, and if so, backs up and retransmits the series of frames beginning with the frame whose sequence number matches the number which the FARM is expecting. See the CCSDS 202.0-B-1 Blue Book for complete specification of the COP-1 protocol.

Output of the command module consists of CCSDS command transfer frames, encapsulated in CLTUs with a Wallops, Universal Space Network (USN), or TDRSS compatible header and trailer. The command module uses the TCP/IP or UDP/IP network protocol.

Telemetry module

The telemetry module accepts CCSDS transfer frames or CCSDS packets from either real-time streams or archive files. These frames or packets may be `wrapped' in TDRSS, Wallops, or USN compatible header and trailers. The telemetry module uses either the TCP/IP or UDP/IP network protocol. When using TCP/IP, the telemetry module can connect to the telemetry source or can listen for a connection from the telemetry source -- the telemetry module can behave as either the client or the server.

Once the telemetry module has received the transfer frames, a packet re-assembler reads the transfer frames and re-assembles the CCSDS source packets, which are made available to downstream processes. The data unpacker then reads the CCSDS source packets, extracts the individual telemetry values, and writes them into the CVT.

Display module

The display module consists primarily of four broad capabilities for displaying telemetry mnemonic values: display pages, sequential prints, plots, and stripcharts. The display module receives current values of mnemonics from the CVT or the datapoint server.

There are two types of display pages: text and graphic. Text pages consist entirely of text, while graphic pages can contain text, gauges, and plots. The graphic pages are Java based. Sequential prints consist of plain text and information about telemetry mnemonics, principally the mnemonic’s value. Plots and stripchart displays are both displays that plot the running values of mnemonics in real-time.

STOL Overview

STOL is the primary interface for controlling the ITOS software. The ITOS dialect of STOL allows ITOS to be controlled interactively or via STOL procs. STOL procs are powerful scripts using if-then-else, do-while, and wait-until control structures. STOL can also access telemetry values and understands all spacecraft commands and telemetry mnemonics in the telemetry and command database. There are over 50 directives in STOL which include the capability to control telemetry acquisition, send spacecraft commands, display telemetry pages, start procs, and all other control aspects of ITOS.

Further information regarding ITOS can be found at http://itos.gsfc.nasa.gov/

Modification for GLAST

Produce frame files that are sorted by time and have no duplicate frames.

2.2.2 Real-Time Automation Monitoring and Control (AMAC)

The Real-time Telemetry and Command Subsystem will be executed on two different workstations, one that will process DAS data received via DAS and the other which will process downlink data received for scheduled SN or GN contacts. Because of differing processing requirements for these two types of data, a separate version of the real-time AMAC software is needed for each.

2.2.2.1 DAS Automation Monitoring and Control (DAS AMAC)

The real-time DAS Automation Monitoring and Control (DAS AMAC) software is based on Swift legacy TDRSS AMAC. It controls all aspects of real-time telemetry processing of data received through the TDRSS DAS. The DAS AMAC determines which STOL directives will be submitted to ITOS instance executing on the ITOS Real-time Alert workstation. The appropriate STOL proc is selected based on entries in the integrated contact schedules and SWSI alert logs. The STOL procs submitted to ITOS contain the directives needed to initiate contact with DAS, read the telemetry stream, get the Burst Alert packets and forward them to BAP, and to close the connection with DAS. ITOS will run continuously during non-scheduled contacts in order to handle burst alerts immediately.

The DAS AMAC parses the ITOS event logs to determine telemetry statistics as well as to detect erroneous conditions. From the telemetry statistics collected, a report is generated in ASCII format that is written to the MOC file server and the Web Server. This report is read, processed and displayed by ITOS and the MOC web interface. Erroneous conditions that are detected by the DAS AMAC will result in paging requests being forwarded to the Virtual Mission Operations Center/Spacecraft Emergency Response System (VMOC/SERS).

The software processes involved in DAS AMAC are illustrated in Figure 2.2.2-1 DAS AMAC Software Flow Diagram.

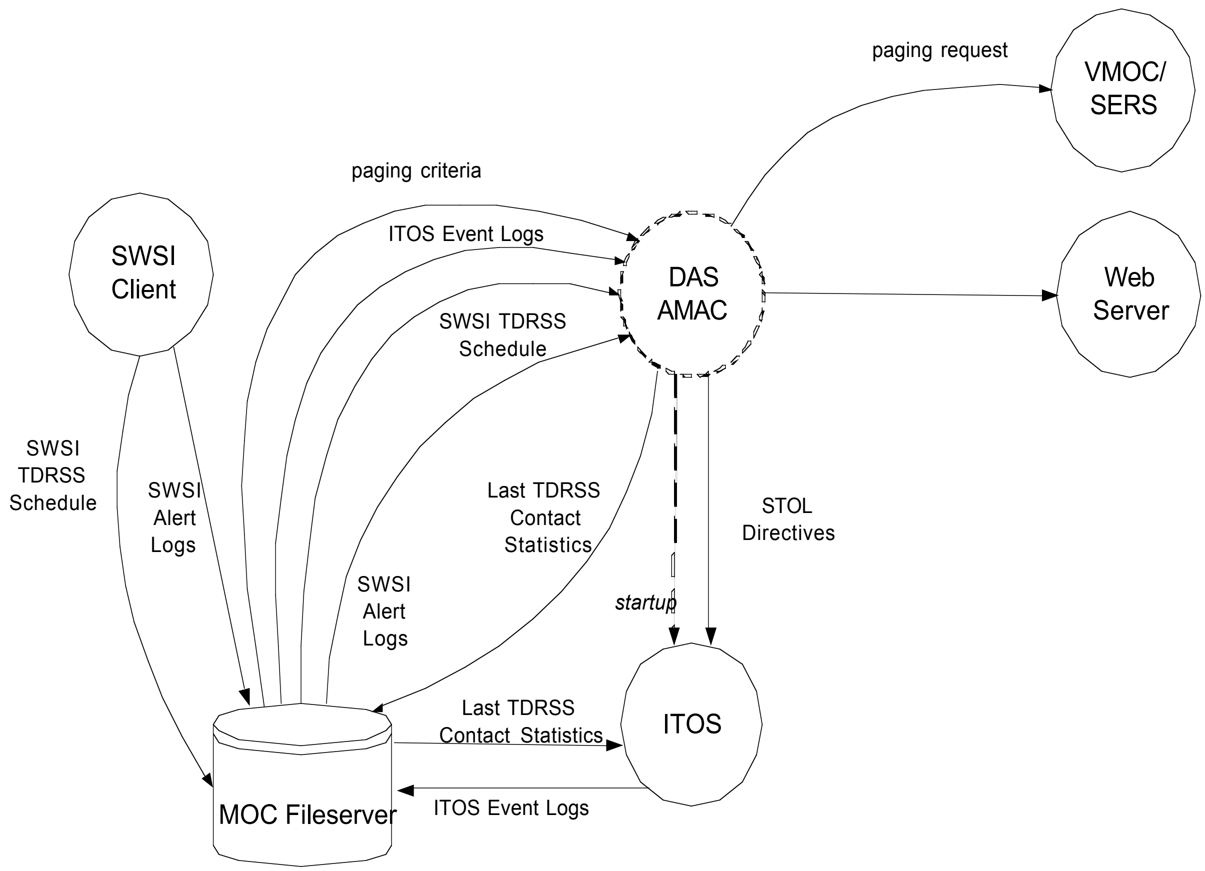


Figure 2.2.2-1 – DAS AMAC Software Flow Diagram

2.2.2.1.1 DAS AMAC Processes

The DAS AMAC software is composed of three processes: DAS AMAC Controller, Event Log & TDRSS Schedule Processor and the SWSI Alert Logs Processor. The DAS Automation Monitoring and Control (DAS AMAC) Process Model is depicted in Figure 2.2.2-2.

<u>DAS AMAC Controller</u>
The DAS AMAC Controller is responsible for starting ITOS, the Event Log & Schedule Processor, and the SWSI Alert Logs Processor. This ordering does not dictate the sequence in which they will be executed.

<u>Event Log & Schedule Processor</u>
The Event Log & Schedule Processor is responsible for the following:

- Reads and processes scheduled contact entries from the Integrated Contact Schedule and according to the scheduled contacts, issues the appropriate STOL directives to ITOS.
  - Issues the STOL directive for DAS AMAC Startup ITOS proc
  - Issues the STOL directive for DAS AMAC Alert Monitoring ITOS proc at the end of each scheduled contact to resume continuous alert monitoring and processing.
  - Issues the STOL directive for DAS AMAC Management ITOS proc at the beginning of each scheduled contact to terminate logs and frame files.
- Read in (or have prior knowledge of) the paging criteria for the current event log session.
- Read in the ITOS Event Logs to determine the status of ITOS and any ongoing ITOS procs as well as collecting segment statistics
- Generate an ASCII file containing the collected segment telemetry statistics. Store the file to the MOC file server and the machine running the Web Server. This file will be read and displayed by ITOS to the ITOS user as well as displayed on the TDRSS Segment Telemetry Statistics web page.
- In the event that an erroneous condition is found in the event logs, send a page request to VMOC/SERS

SWSI Alert Logs Processor

The SWSI Alert Logs Processor is responsible for reading SWSI alert logs and if any erroneous conditions are identified when parsing these logs, to generate a paging request to the VMOC/SERS.

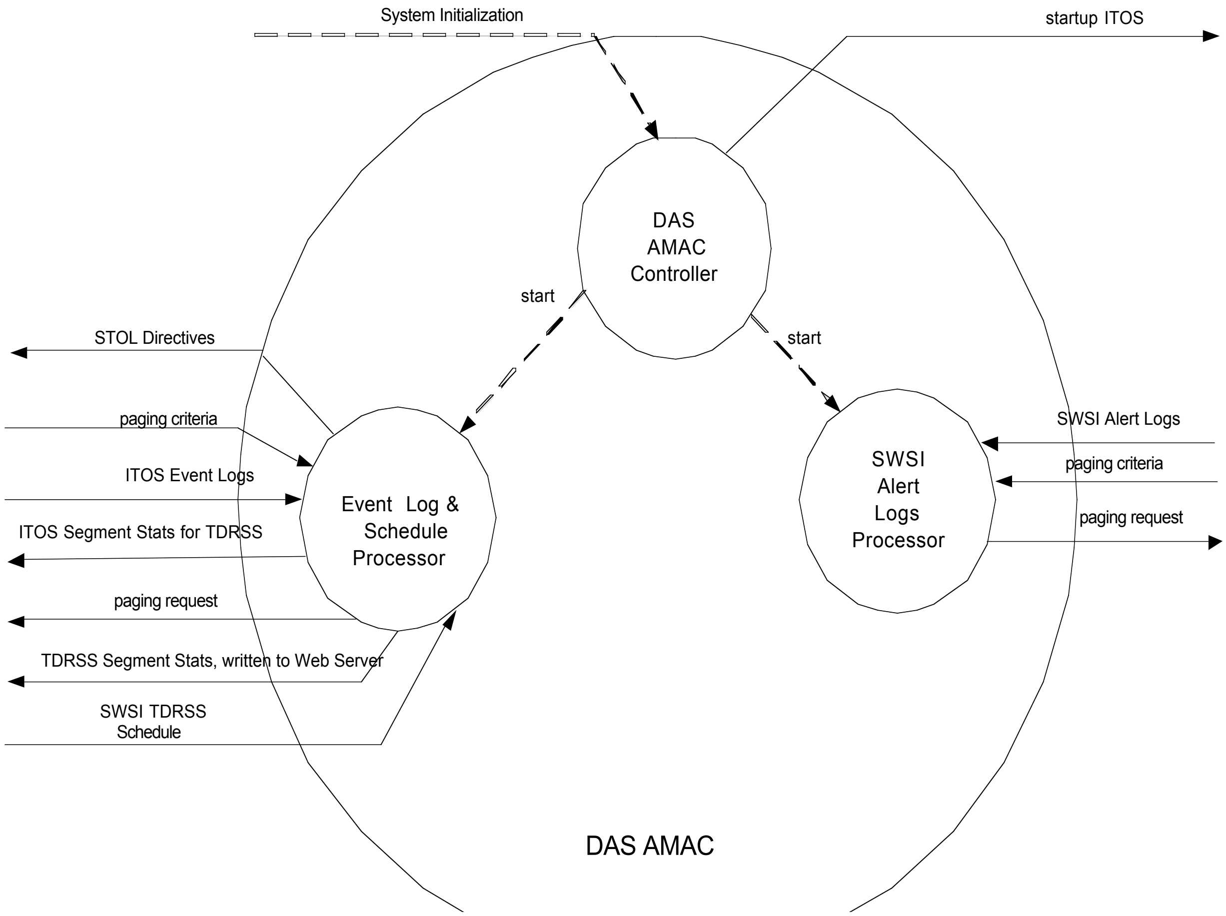


Figure 2.2.2-2 DAS AMAC Process Model

2.2.2.1.2 DAS AMAC Startup ITOS proc

The DAS AMAC Startup ITOS proc that initializes the TDRSS processing performs the following tasks:

- Display the page showing TDRSS status.
- Initialize ITOS telemetry modules.
- Open TDRSS server sockets.
- Open VC1 frame archive.
- Start VC1 frame relay to open server and other external systems if necessary.
- Start configuration monitor checking.
- Start limit violation checking.

2.2.2.1.3 DAS AMAC Management ITOS proc

The DAS AMAC Management ITOS proc performs the following tasks:

- If started at the start of a scheduled contact, closes the old alert frame archive if there is data in it and opens a new frame archive. Then sends old frame archive file to the MOC archive for Level 0 processing.
- Closes the old ITOS event log and opens a new ITOS event log.
- Initiates the event delogger program which will parse and distribute event log to various systems.

2.2.2.1.4 DAS AMAC Alert Monitoring ITOS proc

The DAS AMAC Alert Monitoring ITOS proc performs the following tasks:
- Detect a Burst Alert packet and forward it to the BAP
- Additional processing TBD

Design Issues:
- *The state of ITOS during scheduled contacts is TBD.*

2.2.2.2 Realtime Automation Monitoring and Control (RT AMAC)

The Realtime Automation Monitoring and Control (AMAC) software will control all aspects of real-time telemetry processing for the scheduled TDRSS contacts. The software is based on the Swift Ground Station AMAC. However, for GLAST, ground station passes will not be taken automatically, since they are for backup only. Realtime AMAC starts an instance of ITOS on the real-time telemetry and command system. The Realtime AMAC software is driven by four external inputs:
- ITOS event logs
- Integrated contact schedule
- Manual edits to the contact task schedule
- The Contact task schedule

Throughout all of its automation operations the Realtime AMAC also reads the integrated contact schedule and the contact task schedule. From the information retrieved from these schedules, the appropriate STOL directives are submitted to ITOS.

Realtime AMAC creates the contact task schedule based upon the newest integrated contact schedule. The list of parameters that make up a task schedule record is specified in the Realtime AMAC configuration file. The parameter list can be modified to fit the needs of the Real-time AMAC Contact Proc. The FOT uses the contact task schedule editor to modify the contact and proc parameters for a specific task. The Realtime AMAC then uses the modified task schedule for future contacts. It updates the task schedule with the contact status based on the ITOS event log entries. The contact task schedule is published to the MOC Open Server after each pass.

While ITOS executes the proc, the Realtime AMAC parses the ITOS event logs to determine the status of ITOS and the proc execution. The status of the contact task entry is updated to indicate if the contact is ongoing, completed, or failed. If an erroneous condition is detected, it sends a paging request to SERS based on the paging criteria.

The contact task schedule editor is based on Swift software. When it is started, the editor selects the newest task schedule from either the Realtime AMAC contact task schedule directory or from the Edited Contact Task Schedule directory. The editable fields are specified in a configuration file. The FOT may change the task parameters as necessary. The modified contact task schedule is stored in the Edited contact task schedule directory.

These software components and their associated counterparts are depicted in Figure 2.2.2.2 Realtime AMAC Software Flow Diagram.

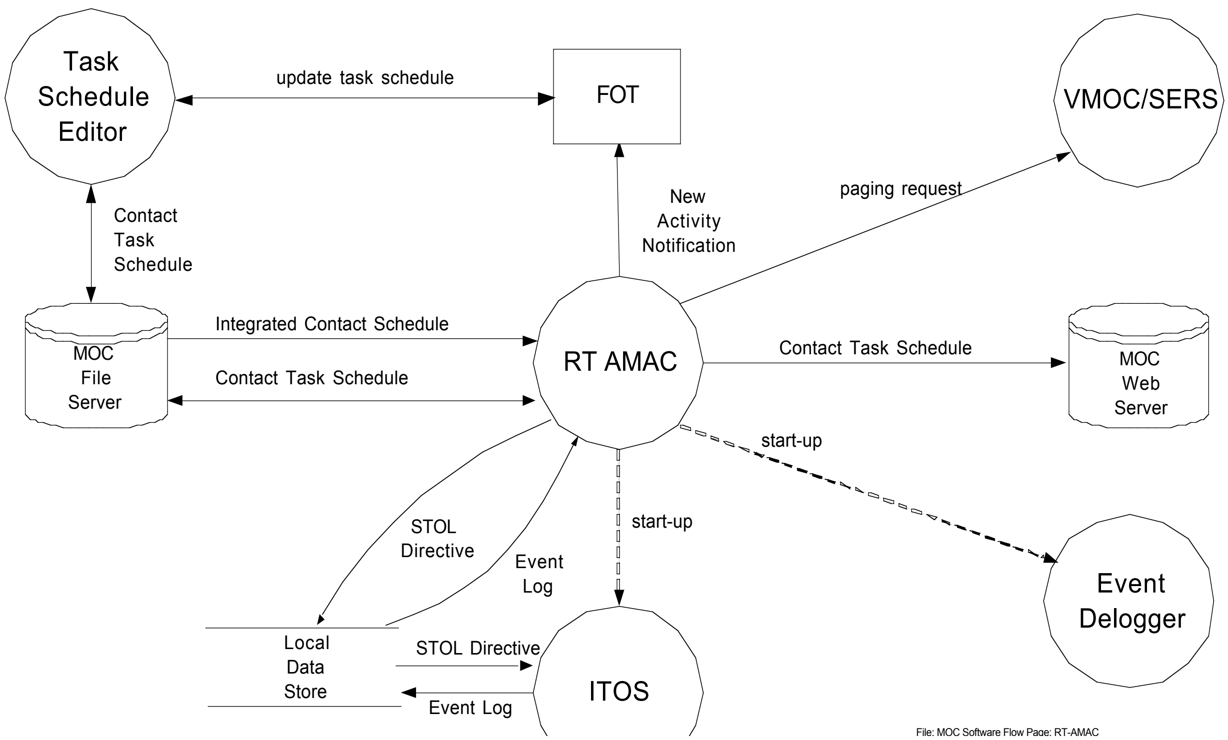


Figure 2.2.2.2 - Realtime AMAC Software Flow Diagram

2.2.2.2.1 Realtime Automation Monitoring and Control (RT AMAC) Processes

The Realtime AMAC software is composed of four components: Realtime AMAC Controller, Contact Schedule Processor, Task Schedule Processor, ITOS Event Log

Monitor and Task Schedule Editor. The Realtime AMAC Process Model is illustrated in Figure 2.2.2.2.1.

Realtime AMAC Controller

The Realtime AMAC Controller is the main loop that sequences the execution of the Contact Schedule Processor, Task Schedule Processor, and ITOS Event Log Monitor. It performs initializations such as opening the AMAC process log and reading the configuration file.

Contact Schedule Processor

The Contact Schedule Processor is responsible for the following:

- Read in the integrated contact schedule and the contact task schedule file. Create or update the contact task schedule file.
- Writes the contact task schedule to the MOC file server and the MOC web server.
- Create an ITOS display page file with the content of the task schedule.
- If it is the first time that the contact task schedule has been generated or additions have been made to the contact task schedule, a child process is launched to display a pop up window to the FOT listing the contact task schedule additions in pass sequence order.

Task Schedule Processor

The Task Schedule Processor is responsible for the following:

- Reads the begin time of the first contact task schedule record to see if it is within the range of the ITOS prep time (the amount of time before the pass is to start that ITOS needs to be started). If it is NOT, the RT AMAC returns to looking for updates to the ICS and the process begins again.
- If the begin time of the first contact task schedule record is within the range of the ITOS prep time, the AMAC initiates ITOS with the Realtime AMAC proc, passing as proc parameters all necessary values contained in the contact task schedule record.
- Starts the Event Delogger passing it the most recently created event log.
- Sends a page request to VMOC/SERS when any condition occurs that prevents automatic telemetry processing.

ITOS Event Log Monitor

The ITOS Event Log Monitor does the following:

- First time only, reads the AMAC configuration file to determine the ITOS events that the AMAC can take action on.
- Reads in the ITOS Event Logs to determine the status of ITOS and any ongoing ITOS procs.
- If the event is “stopped due to errors” : This is a signal from ITOS that it has encountered some problems and needs to be restarted.

- If the event is “Pass Processing Status” : This is a pass status message from ITOS indicating the new status of the pass. This new status is used to update the task schedule entry for the respective pass.
- Sends a page request to VMOC/SERS when any condition occurs that prevents automatic telemetry processing.

Paging criteria includes but is not limited to the following:

- Contact task schedule could not be created or updated.
- Event Delogger could NOT be invoked.
- ITOS failed to respond within the time period set for this pass.
- ITOS Event Log directory could NOT be detected OR is not readable.
- ITOS STOL FIFO could NOT be detected.
- Integrated Contact Schedule files could not be located.
- A new ICS was received but the FOT could not be notified of the additions because no one was logged into the console.

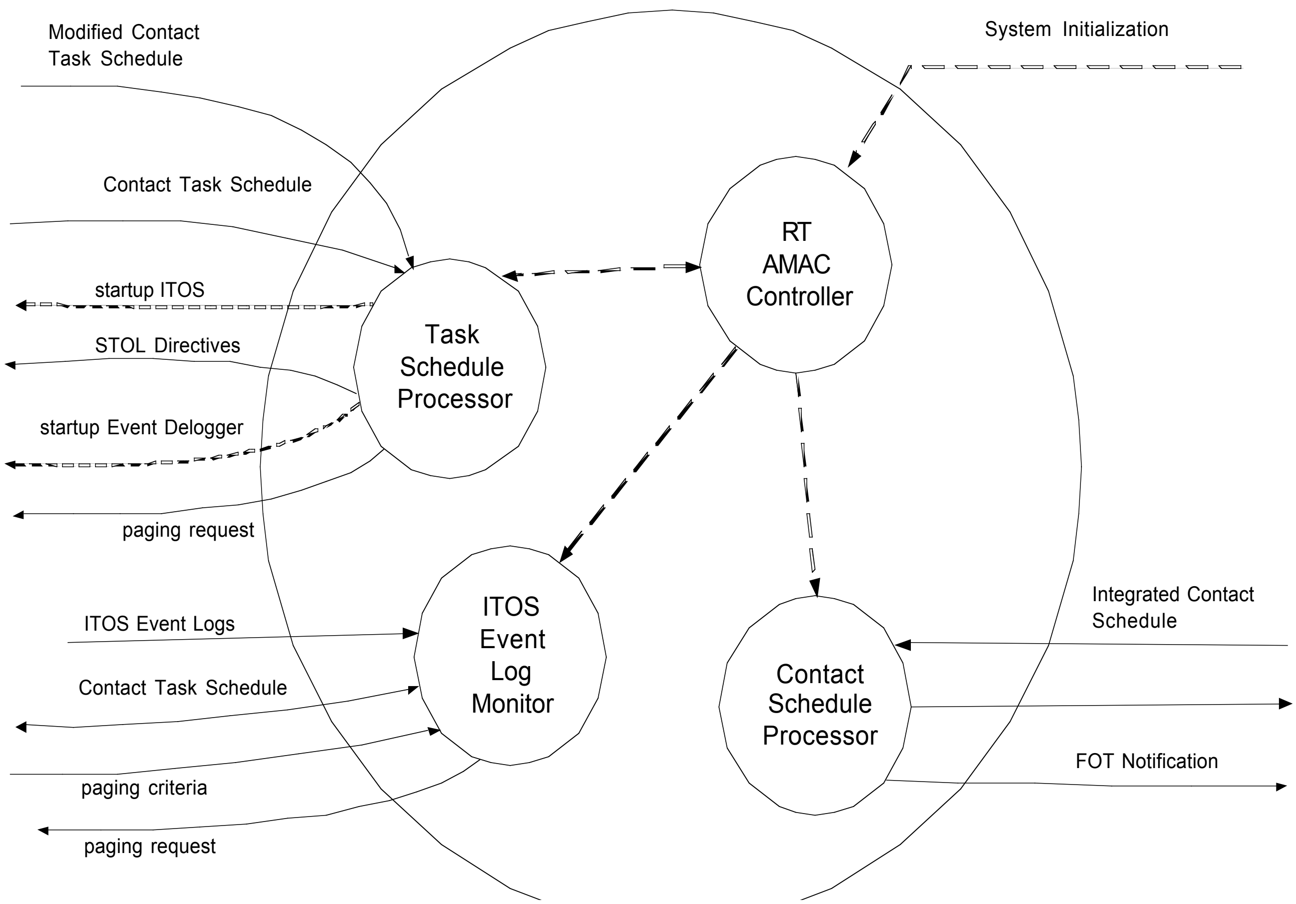


Figure 2.2.2.2.1 - Realtime AMAC Process Model

2.2.2.2.2 Realtime AMAC Real-time Contact ITOS proc

The Realtime ITOS proc that performs the scheduled TDRSS contacts does this:

- Validate input parameters: AOS time and LOS time
- Open event log file
- Initialize ITOS telemetry and command modules.
- Open telemetry and command server sockets.
- Open VC0 frame archive.
- Start sequential prints
- Start VC0 frame relay to open server and other external systems if necessary.
- Open telemetry pages
- Start configuration monitor checking.
- Turn on limit violation checking

- Wait for LOS
- Close VC0 frame archive.
- Close telemetry and command server sockets.
- Write the general pass status to event log.

2.2.3 Task Schedule Editor

The Task Schedule Editor is invoked by the user when a contact task schedule entry must be modified. The editor automatically selects the newest task schedule from either the AMAC Contact Task Schedule directory or from the Edited Contact Task Schedule directory. The editable fields along with each field's validation criteria are specified in a configuration file. The FOT selects the desired entry from a list and may delete the entry or change the task parameters as necessary. New entries may not be added using the editor, since they must be based on scheduled contacts. The FOT then saves the modified contact task schedule. The modified file is stored in the Edited Contact Task Schedule directory. If the AMAC has created a new task schedule during the editing session, the editor requires the user to load the new task schedule, discarding any edits to the old schedule.

2.2.4 Real-time Event Delogger

The Event Delogger subsystem is used to create automated reports from the ITOS event logs. It is based on Swift-legacy software. Reports are a specified subset of the event logs. The Event Delogger also passes the ITOS event logs to the VMOC/SERS software for alert processing.

The Event Delogger is scheduled to run every 5 minutes in order to process the results of demand access TDRSS contacts. Limit violations, configuration violations, and real-time commands are extracted from the ITOS event logs into separate report files that are stored on the MOC fileserver and optionally sent to the printer. The ITOS event logs are forwarded to the VMOC/SERS for alert processing. The Event Delogger also moves the ITOS event logs to the MOC fileserver for archiving.

2.3 Mission Monitoring & Offline Processing

The Mission Monitoring & Offline Processing subsystem is composed of seven software components: Data Archiver, Frame Accounting Software, Event Delogger, Offline Automation Management and Control (AMAC), Timeline Monitor, Virtual Missions Operations Center/Spacecraft Emergency Response System (VMOC/SERS) and ITOS. These subsystems are depicted below in Figure 2.3 – Mission Monitoring & Offline Processing Software Flow Diagram and are described in more detail in the subsections that follow.

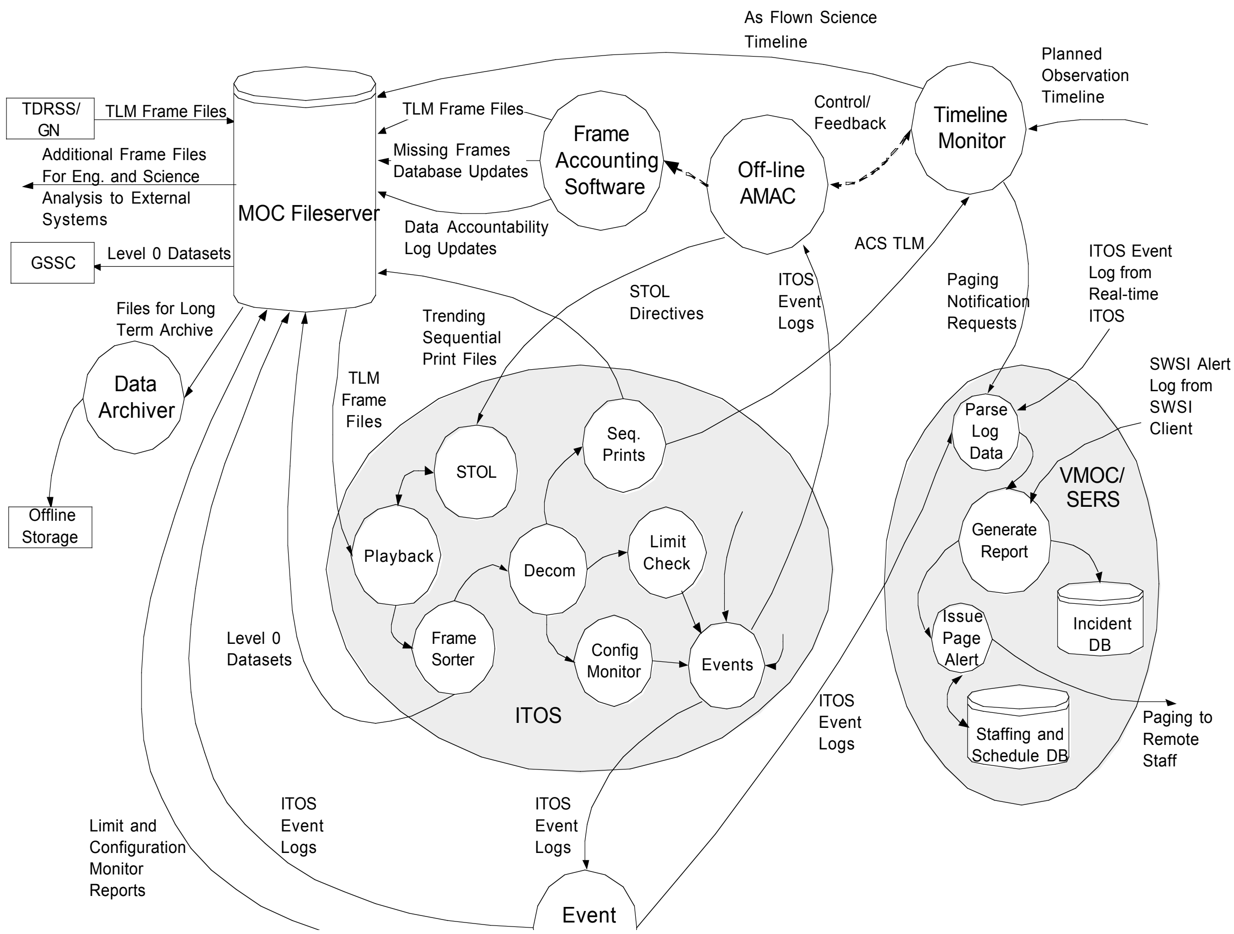


Figure 2.3 – Mission Monitoring & Offline Processing Software Flow Diagram

2.3.1 Data Archiver

The Data Archiver component is implemented with COTS software and is used to copy data files to offline media (e.g. CD, tape, etc.). On a periodic basis, the Data Archiver retrieves and archives CCSDS Frame files, Integrated Observatory Timeline files and Sequential Print files from the MOC Fileserver. These files are compressed before being stored. The archived data is maintained for the life of the mission.

2.3.2 Frame Accounting Software

The Frame Accounting Software is Swift legacy software that is responsible for processing telemetry files, collecting missing frames and frame accountability statistics from those telemetry files, and updating the respective data stores on the MOC file server.

The Frame Accounting Software has two main processes, the frgap process for determining and logging frame gaps and the frquery process for determining and logging the overall quality statistics. These processes are depicted below in Figure 2.3.2.1 Frame Accounting Software Process Model.

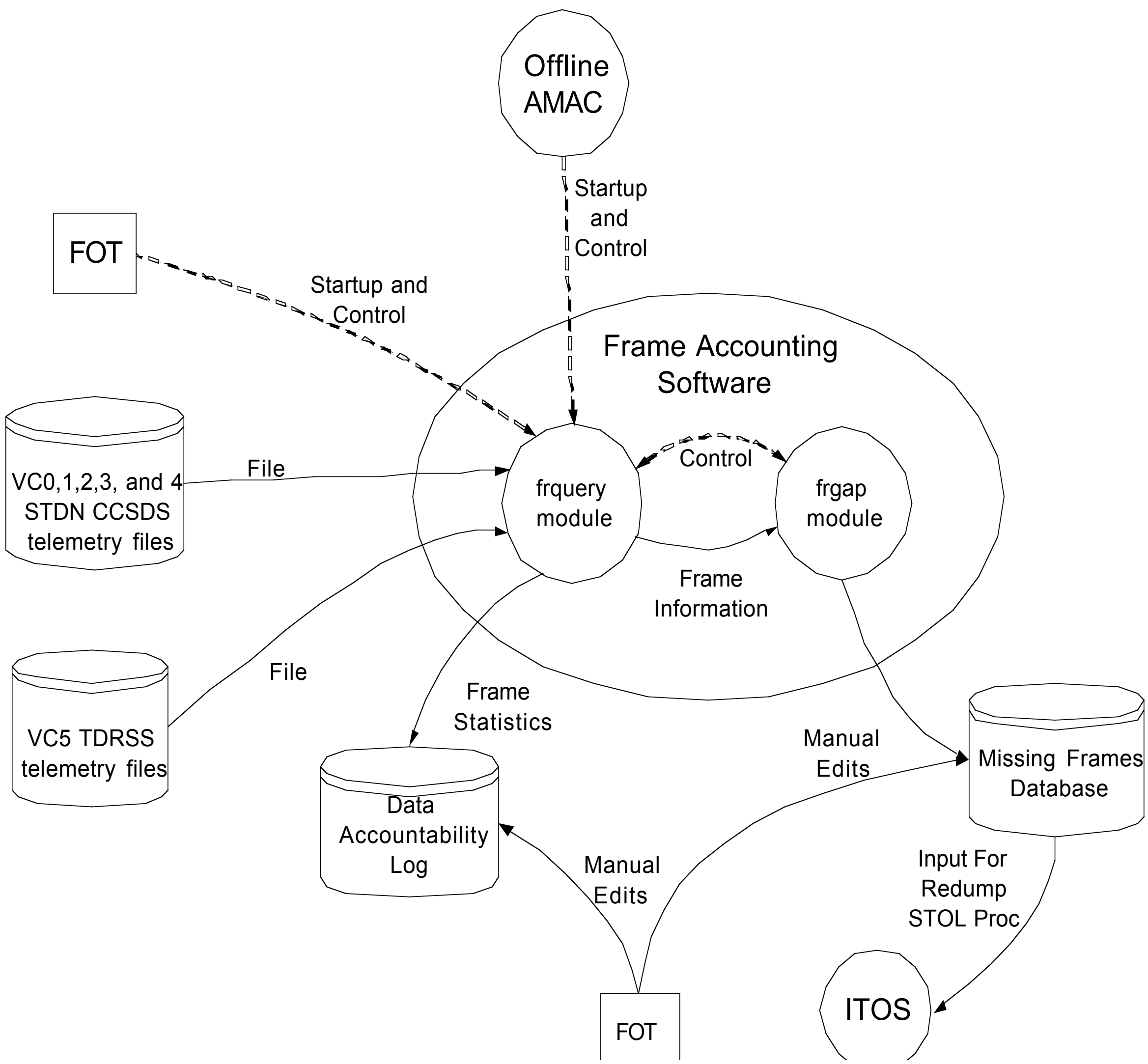


Figure 2.3.2.1 – Frame Accounting Software Process Model

<u>Frquery</u>

Either the Flight Operations Team (FOT) or the Offline AMAC can initiate the frquery process. The frquery process reads each telemetry file as it is archived from the station to the MOC file server. Each telemetry file may contain several Virtual Channels (VCs). The frquery process determines the following data quality statistics for each VC:

- Total frames
- Total good frames
- Total frames with recoverable Reed-Solomon (RS) errors
- Total frames with unrecoverable RS errors
- Sequence errors
- First valid frame time
- Last valid frame time
- Percent of unrecoverable RS errors with respect to total frames.

The logging statistics collected will be tab delimited and written to a statistics report. The statistics report is stored on the MOC Fileserver and sent to the Web Server.

<u>Frgap</u>
Either the Flight Operations Team (FOT) or the Offline AMAC can initiate the frgap process. The responsibilities of this process are to identify sequence gaps in a virtual channel, and to log those missing frames to the Missing Frames Database.

Telemetry files containing one or more VCs will be processed, looking for sequence gaps by checking the frame sequence counter while also performing other Reed-Solomon quality checks and frame header verification. If the frgap process detects non-linearity in the sequence number, it logs this sequence gap to the Missing Frames Database along with the time stamp from the frame which preceded the frame gap as well as the time stamp from the frame which succeeded the frame gap. The Missing Frames Database is a cumulative record of the gaps detected over all frame files processed by frgap. The sequence gaps detected in an individual file are written to the missing frames report. The FOT uses this sequence gap information to determine which segments of data need to be re-dumped. The missing frames report and the Missing Frames Database are stored on the MOC Fileserver and sent to the Web Server.

The frgap program uses the transfer frame replay flag to distinguish between normally dumped data and redumped data. Typically, the redumped frames are used to remove gaps from the Missing Frames Database. However, if the redumped frames have not been processed before, then they will be processed just like normal playback frames. Any gaps will be added to the Missing Frames Database.

Frall
The Frall process combines the functionality of the frquery and frgap processes. It produces statistics report, the missing frames report, and the missing frames database.

2.3.3 Event Delogger

The Event Delogger is Swift legacy software used to create automated reports from the ITOS event logs. Reports are a specified subset of the event logs. The Event Delogger also passes the ITOS event logs to the VMOC/SERS software for alert processing.

The Event Delogger is scheduled to run following the playback of housekeeping data by the Offline ITOS system. Limit violations, configuration violations, and real-time commands will be extracted from the ITOS event logs into separate report files that will be stored on the MOC fileserver and optionally sent to the printer. The ITOS event logs are forwarded to the VMOC/SERS for alert processing. The Event Delogger also moves the ITOS event logs to the MOC fileserver for archiving in the $FS_RPT_EVT_LOG_OFFLINE directory

2.3.4 Offline Automation Monitoring and Control (Offline AMAC)

The Offline Automation Monitoring and Control (Offline AMAC) software shall control all aspects of the offline telemetry.

After initialization and start-up, the Offline AMAC queries the MOC file server to see if there are telemetry files to process. If there are no files to process, it continues to poll the file server. If there are telemetry files to process they are queued and Offline processing begins to process one telemetry file at a time from the queue.

First, the Offline AMAC initiates the Frame Accounting Software to analyze data quality and contents of the received telemetry files. The Frame Accounting Software logs its results to various output logs and distributes these results to the Web Server for display on the GLAST MOC web site. Then an instance of ITOS is started on the ITOS Offline workstation. This instance of ITOS executes the Offline AMAC ITOS Processing Proc for startup and processing of the telemetry file passed to it. All telemetry files require Level 0 dataset processing while only VC?? data will require offline ITOS processing.

The Offline AMAC parses the ITOS event logs to detect completion of telemetry file processing as well as erroneous conditions. Upon detection of telemetry file processing completion, the Offline AMAC starts the Timeline Monitor only if the telemetry file that just completed processing was from VC??. It initiates the Event Delogger program which will parse and distribute event log to various systems, given the path and file name of the event log. Then, the Offline AMAC builds and sends the Level 0 signal file to the GSSC. The

Offline AMAC checks its queue of telemetry files and if the queue is not empty, it will select another telemetry file from the queue. The Offline processing sequence of Frame Accounting, ITOS and Timeline Monitor is repeated until the queue of telemetry files is exhausted. When the list of telemetry files are processed, and if observatory housekeeping data was processed, Offline AMAC distributes the Level 0 Datasets to MOC Open file server, then builds and sends a Level 0 signal file to the GSSC. Finally, the Offline AMAC builds and sends a pass complete signal to the GSSC.

Erroneous conditions that are caught by the Offline AMAC result in paging requests and/or the restarting of ITOS. Paging requests are forwarded to the VMOC/SERS. ITOS is restarted and the telemetry file is reprocessed if the erroneous condition warrants such action.

Offline AMAC Processing ITOS Proc

The ITOS proc for offline ITOS processing shall perform the following tasks:

- Start telemetry file playback
- Start CCSDS packet extraction to Level 0 Datasets.
- DTAS sequential print file data capture on VC1 and VC6 playbacks.
- ITOS sequential print data capture of observatory state telemetry
- ITOS event log capture
- Configuration monitor starting and processing
- Limit violation checks
- Indicate through event log that VC1 or VC6 processing has completed. This will be indicated in the event log through a "PROCESSING COMPLETE" record entry in the event log.
- Distribution of DTAS sequential print files to DTAS and trending archive.
- Distribution of observatory state sequential prints to MOC archive and the Timeline Monitor staging directory.

These software components and their associated counterparts are depicted below in Figure 2.3.4 - Offline Automation Monitoring and Control Software Flow Diagram. For a more detailed description of the Offline AMAC, see the "Offline AMAC Detailed Design/Programmers Guide".

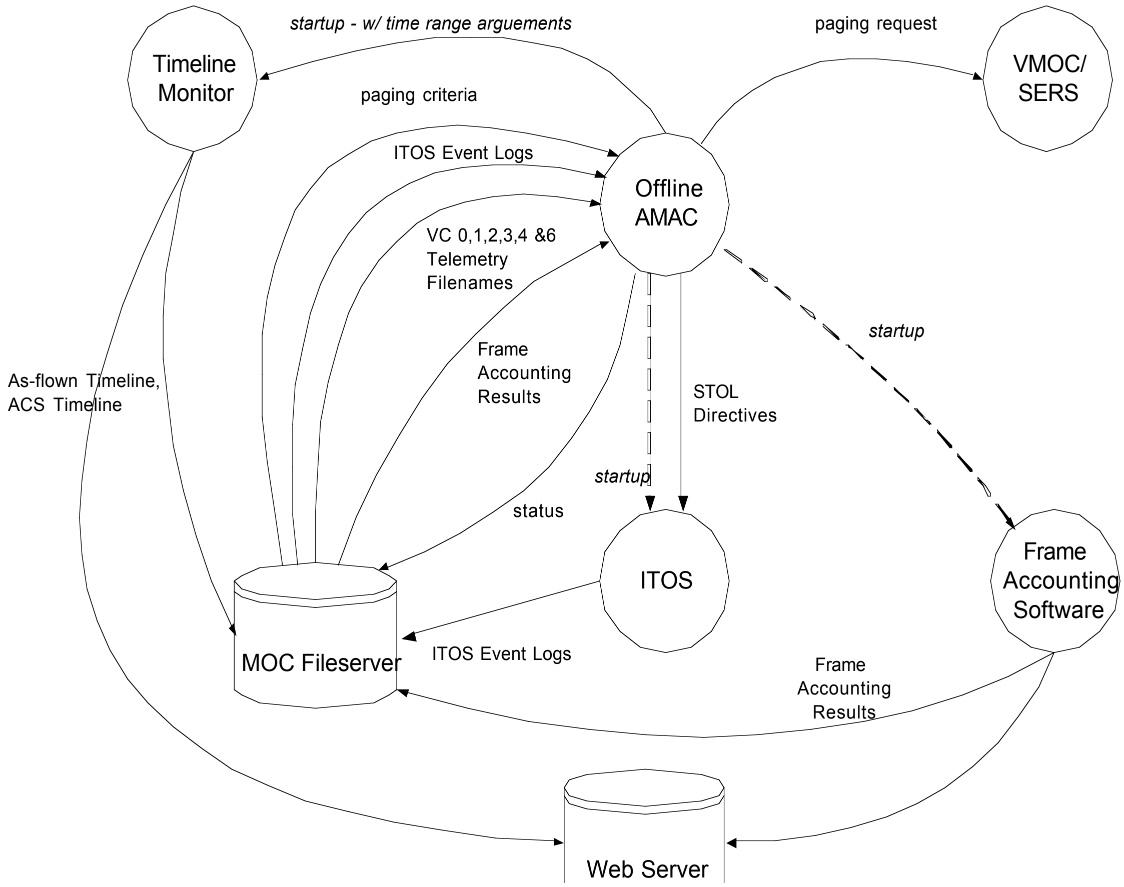


Figure 2.3.4 – Offline Automation Monitoring and Control Software Flow Diagram

2.3.5 Timeline Monitor

The Timeline Monitor is based upon Swift-legacy software, but must be heavily modified for the GLAST observing profile. This component is responsible for verifying that the pre-planned science timeline is proceeding onboard the spacecraft as planned. Any deviations from the pre-planned science timeline is noted as normal autonomous re-pointing, survey modes, or as a timeline anomaly. These interruptions or anomalies are noted via a report, email, or paging request. The timeline validation software generates an As-Flown Timeline from the information gathered.

<u>As-Flown Science Timeline Generation</u>

The As-flown Science Timeline generation is designed to be as automated as possible, however the software may not be able to automatically validate the timeline in all cases. For those few cases, manual changes to the as-flown timeline are expected. The As-Flown Timeline format is defined in the *GLAST Operations Data Products ICD*. Key capabilities for the as-flown timeline generation are:

- Software operates interactively or via automation control.
- Identify, select, and sort ITOS sequential print data for time range needed.
- Identify any significant gaps in observatory state telemetry data. Small gaps less than one minute will not be consequential to timeline validation. If there are significant gaps in the spacecraft attitude data, the software will be unable to validate the timeline.
- Select and read the planned observation timeline and identify pointed targets and their associated attitude, instrument configuration, time, etc. The planned observation timeline is a list of scheduled pointings and instrument mode changes that were or will be commanded.
- Compare PT parameters and observatory state values to determine if PTs are occurring when planned.
- Create an As-Flown Timeline based on PT validation and identified Autonomous Re-points (AR) and real-time ToO timeline deviations. Indicate in the As-Flown Timeline where PT targets could not be validated because of missing telemetry data. TBD logging of Report Survey Mode and Earth-limb tracing mode, etc.
- Timeline validation algorithms handle missing telemetry data without causing false modifications to the pre-planned timeline when creating the as-flown timeline.

For a more detailed description of the As-Flown Timeline Generation module, see the "Timeline Monitor Detailed Design/Programmers Guide".

2.3.6 VMOC/SERS

The Virtual Mission Operations Center/Spacecraft Emergency Response System (VMOC/SERS) is a GOTS package which will automate anomaly notification at the MOC. VMOC/SERS will receive anomaly messages from the Timeline Monitor, and other MOC components. These external clients perform limit checking and anomaly detection. VMOC/SERS will also parse ITOS event log files for anomalous conditions after scheduled passes.

VMOC/SERS will rely upon an external paging service (such as SkyTel) for two-way paging capability. MOC operational personnel will be equipped with two-way pagers. Upon receiving a VMOC/SERS page, a remote staff member may acknowledge the page or defer it to another staff member.

VMOC/SERS inputs include ITOS sequential print files, and alert messages from the Timeline Monitor. VMOC/SERS parses these inputs for observatory anomalies, GRBs, and timeline execution anomalies, respectively. Upon detecting an anomaly, VMOC/SERS queries internal schedule and staffing databases to generate a page request. SERS issues this page request to the appropriate operations personnel (e.g., paging thermal personnel for thermal anomalies) via an external paging service.

Upon receiving a page, MOC personnel can respond to VMOC/SERS via two-way pagers. Personnel can accept responsibility for the page or defer responsibility to other personnel. VMOC/SERS listens for operator responses. If a paged operator defers (or fails to respond at all within a set interval), VMOC/SERS will "escalate" its paging functions. VMOC/SERS will make further database queries and issue further pages until it receives an affirmative response from an operator. In addition to pages, VMOC/SERS can issue alert notifications to operations personnel via e-mail.

VMOC/SERS will store a text summary of each anomaly in an internal event database. This database will be accessible to local and remote operators.

VMOC/SERS will reside on redundant workstations at the MOC with watchdog/handshaking routines between each workstation. The VMOC/SERS provides facilities to maintain staffing and schedule databases to reflect the current operations configuration. Changes in staffing, operator responsibility, pager numbers, etc. must be executed as needed.

2.3.7 ITOS

See Section 2.2.1 Integrated Test and Operations System (ITOS) for a description of the ITOS subsystem. This is a separate instance of the ITOS software used in the real-time processing but in an offline mode. In the offline mode, the ITOS system processes the telemetry frame files and generates Level-0-processed data sets, trending sequential print files and event logs.

A Level 0 data set is created from a single virtual channel telemetry frame file. A data set will consist of either a set of files, each one containing ITOS packet annotation headers and the CCSDS telemetry source packets for a single APID, or one file that contains all headers and packets for all APIDs. These files are constructed "as received" from the spacecraft and are not time ordered or duplicate removed. It is expected that the packets will be in time order for the most part, but there will be packets from re-dump data that will be out of time sequence. The final Level 0 data set file contains the frames in time order and without duplicate frames.

In addition to the Level 0 data set files, ITOS it will also provide statistics and gap information on each data set. The packet accounting information will be generated from the ITOS annotation header and the packet times in the CCSDS secondary header. ITOS will also identify gaps in the packet sequence count and estimate the number missing packets assuming no more than one roll over of the sequence counter for non consecutive counts. (Some APIDs may be identified later that consistently have frequent sequence count gaps. There will be a method to be able to exclude selected APIDs from gap accounting.)

The packet accounting report file will contain three sections. The sections will contain: summary information about the entire data set, summary information for each APID file, and a missing packet list for each APID. For data sets that consist of only one file with all APID packets combined, the report will only contain the first section that contains the summary information about the entire file. The packet accounting report will be named in a way to identify it with the virtual channel frame file from which the data set was created.

The summary data collected for the entire data set will contain:

Data set size in bytes
Time of data set creation
Total number of APIDs
Total number of packets
Total number of packets from frames without RS enabled
Total number of packets from frames with uncorrectable RS error detected
Total number of packets from frames with CRC error
Total number of packet sequence errors
Total number of missing packets
Total number of incomplete packets
Earliest packet time
Latest packet time

The summary data collected for each APID file will contain:

APID number
File size in bytes
Number of packets
Number of packets from frames without RS enabled
Number of packets from frames with uncorrectable RS error detected
Number of packets from frames with CRC error
Number of packet sequence errors
Number of missing packets
Number of incomplete packets
Time of 1st packet
Sequence number of 1st packet
Time of last packet
Sequence number of last packet
Earliest packet time
Latest packet time

The missing packet list for each APID will contain:

APID number

For each packet sequence error:

Location (byte offset) of packet with sequence error
Number of packets missing
Sequence number of packet
Sequence number of previous packet
Packet time
Previous packet time

2.4 MOC Mission Planning & Scheduling

The MOC Mission Planning and Scheduling system is composed of six software components: Satellite Tool Kit (STK), STK Automation Software, Science Input Processor, Contact Schedule Muxer, MPS, and TDRSS Scheduling. These components are depicted below in Figure 2.4 – MOC Mission Planning & Scheduling Software Flow Diagram and are described in more detail in the subsections that follow.

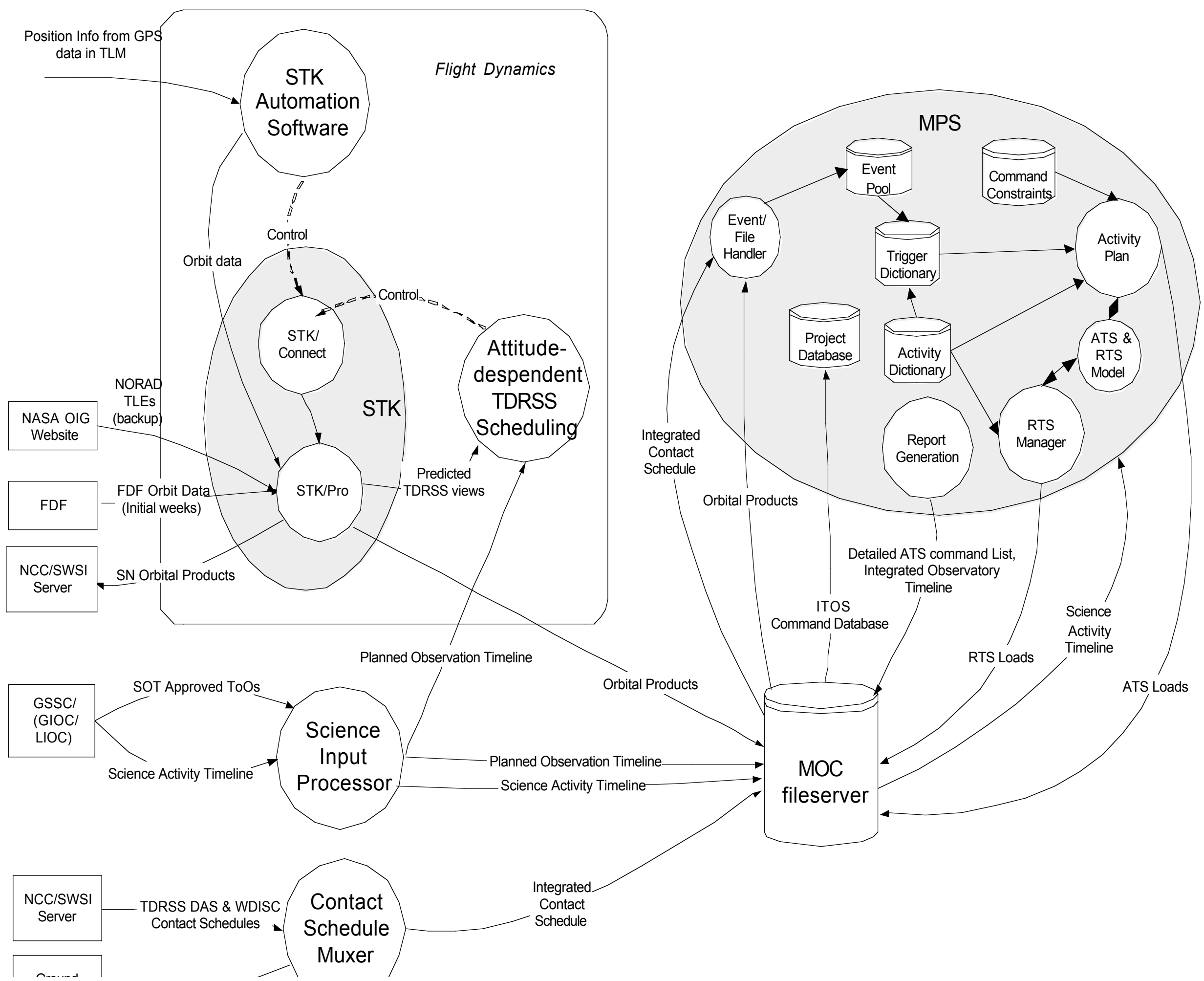


Figure 2.4 – Mission Planning & Scheduling Software Flow Diagram

2.4.1 Satellite Tool Kit (STK)

Satellite Tool Kit is an industry standard COTS suite which provides numerous graphical and analytical orbit management capabilities. The MOC shall use the STK/Professional (STK/PRO) module for orbit determination and anomaly investigation. The STK/Connect module "front-end" enables automated data transfer and processing.

Given the spacecraft position data, STK/PRO produces a spacecraft ephemeris covering the next 12 days. STK/PRO also generates TDRSS ephemerides, TDRSS and station view period reports and orbit event reports. These report files are distributed to MPS and GSSC to assist with the mission planning process. These entities are noted in Figure 2.4 as "Orbital Products".

Control and monitoring of STK will be performed by the STK Automation Software, which is described in more detail in Section 2.4.2.

2.4.2 STK Automation Software

The STK Automation Software is custom software that interfaces with STK/Connect to control STK report generation. It processes the spacecraft's GPS location data from the telemetry to generate orbital products for mission planning. As a backup the spacecraft TLE can also be obtained from the NASA OIG website or CelesTrak website and used in place of the GPS data.

The STK Automation Software sends an ephemeris generation command to STK/PRO to create a 12-day ephemeris for the spacecraft and for each TDRS. These products are used for science planning and TDRSS contact scheduling. STK Automation initiates the transfer of the planning products to the MOC Web Server.

The STK Automation Software then sends an orbit-propagation command to STK/PRO, instructing it to generate TDRSS view period and orbit event files. View period and orbit event files will be supplied to the MPS.

Updated IIRV data is required for TDRSS services. STK Automation Software will also control the generation of a GLAST ephemeris covering approximately 96 hours. STK Automation Software will convert this ephemeris to IIRV (Improved Inter-Range Vector) format and make the IIRV available on the MOC File Server for retrieval by the DSMC/SWSI server.

The STK Automation Software data flow is depicted below in Figure 2.4.2 – STK Automation Software Data Flow Diagram.

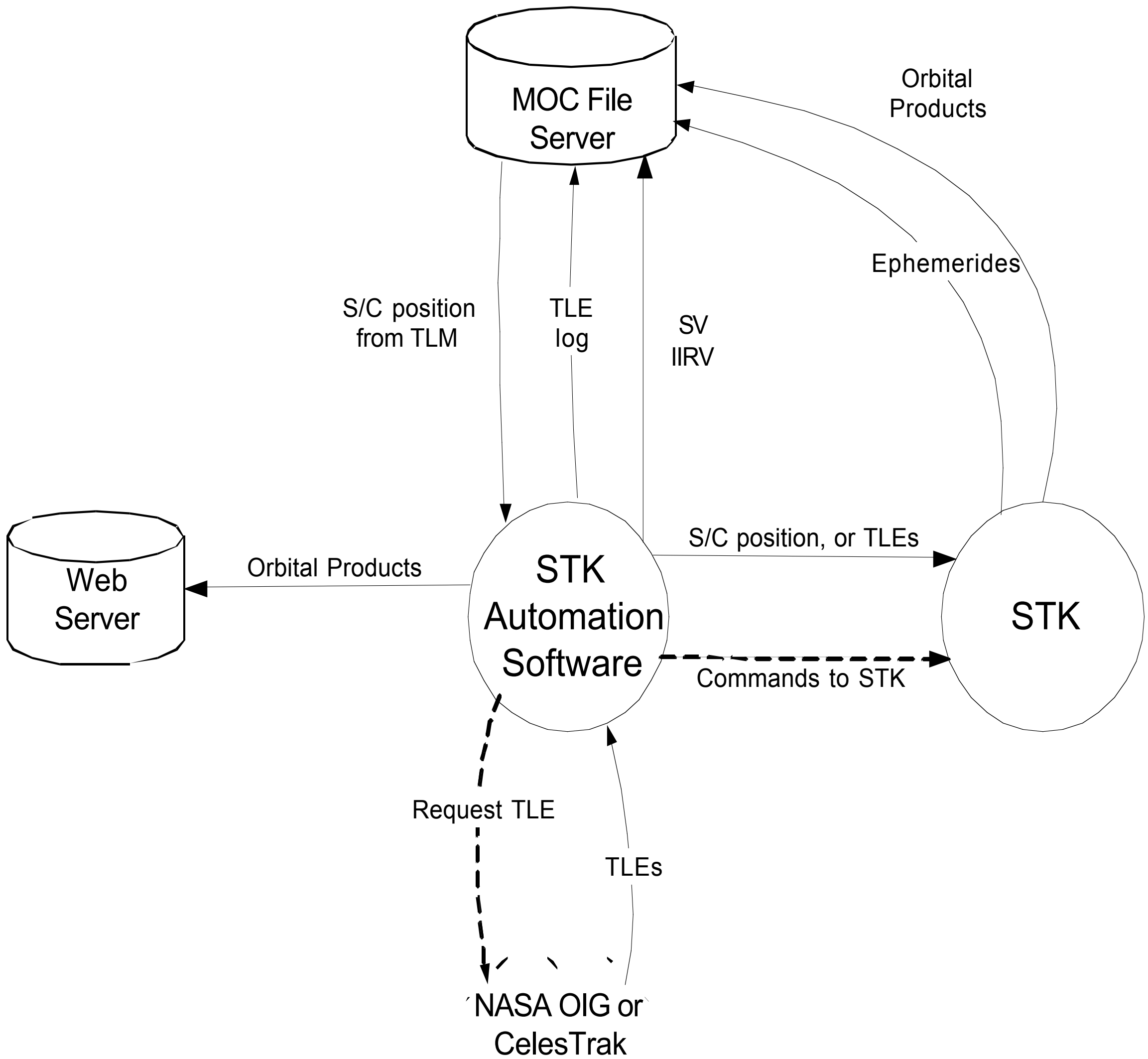


Figure 2.4.2 – STK Automation Software Flow Diagram

2.4.3 Science Input Processor

The Science Input Processor is custom software that will be responsible for processing and distributing files received from the GSSC and, as a contingency, from the IOCs. The Science Input Processor performs the appropriate processing depending on the file type. The commands in the science command timeline are used to create the corresponding planned observation timeline file. (This file is used by STK Automation and the Timeline Monitor.) This software also moves the files to the appropriate directory in the MOC Fileserver, initiates the transfer of the files to the Open Network. The Science Input Processor is depicted below in Figure 2.4.3 – Science Input Processor Software Flow Diagram.

Design Issues:

1. Actual file transfer protocol will slightly modify how SIP is initiated but not the processing. Polling is done with a cron task.

2. Depending on how the ToO Orders are sent, this process could notify the FOT of the arrival of ToO Orders, and maybe even package them into PROCs. (Just an idea, not a requirement) Another option is to have the GSSC email ToO requests directly to SERS. SERS would then email an acknowledgement of receipt back to the GSSC. It would also initiate pages or emails to the FOT.

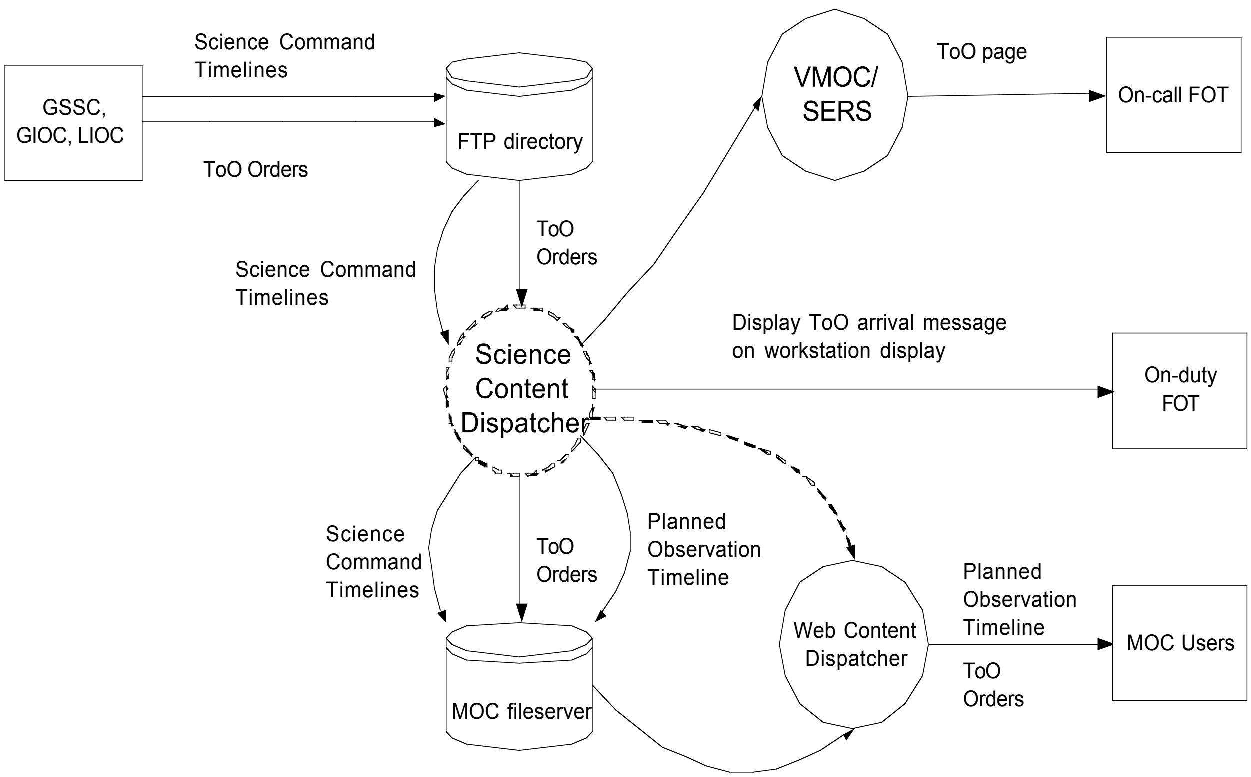


Figure 2.4.3 – Science Input Processor Software Flow Diagram

2.4.4 Contact Schedule Muxer

The Contact Schedule Muxer is Swift-legacy software that is responsible for integrating TDRSS and ground station contact schedules into a single file and consistent format in time order. This component shields the rest of the MOC from changes in external schedule formats. The ground stations transfer their contact schedule files to the MOC using FTP. Two TDRSS contact schedules (WDISC and DAS) are provided to the MOC via the SWSI interface. Once the Contact Schedule Muxer has merged the schedules into the integrated contact schedule, it writes the file to the MOC fileserver for use by other subsystems. The Contact Schedule Muxer is depicted in Figure 2.4.4 – Contact Schedule Muxer Software Flow Diagram.

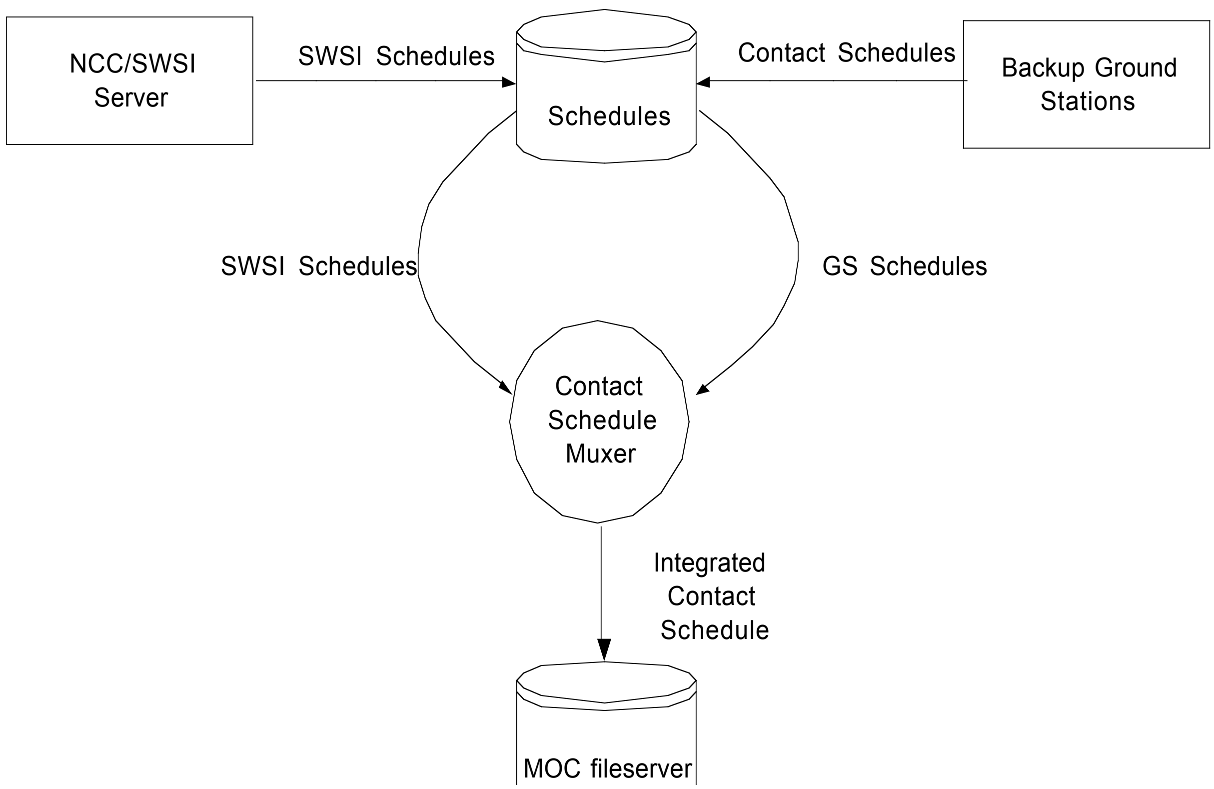


Figure 2.4.4 – Contact Schedule Muxer Software Flow Diagram

2.4.5 Mission Planning System (MPS)

The Mission Planning System (MPS) is GOTS software that enables the FOT to prepare commands that are stored and executed onboard the spacecraft. Using the MPS, the FOT creates and manages an activity plan that is derived from orbital events, activities, triggers, science timeline events and user input. The GLAST MPS is based on the Swift mission version of MPS which already has STK and ITOS interfaces and runs on a Sun Solaris platform. The modifications needed for the GLAST mission-specific requirements are noted in the MPS Organization section below.

In order to fully understand the command load planning process some definitions of terminology are in order.

Activity – a user created sequence of commands and activities that are grouped together and manipulated as a unit. Each command or activity has a relative time tag.

Activity Plan – a list of absolute time tagged activities and commands planned for a several day period. These commands and activities can be triggered (scheduled automatically by events) or input directly by the FOT.

Activity Plan Period – a segment of the activity plan used to define and generate a particular ATS load. The start time and duration of the segment is defined by the user.

Orbital Events – absolute time tagged events related to the operation of the spacecraft (pass events, eclipses, ascending/descending nodes, apogee/perigee) ingested from the orbit products and contact schedules.

Science Timeline Events – a list of absolute time tagged events related to science observations.

Trigger – user-defined logical conditions associated with one or more activities and/or commands. When the conditions are satisfied by an event, the trigger is fired causing the associated activities and commands to be scheduled in the activity plan.

Relative Time Sequence Load – a relative time sequence (RTS) load is used to place certain sets of commands into special areas of spacecraft memory called RTS tables. The command execution times are given as a delta time relative to the time the RTS table is started or to the previously executed command. Execution of the commands in an RTS table can be started by a real-time command or a command from the ATS buffer.

Absolute Time Sequence Load – an absolute time sequence (ATS) load is used to place commands in one of the onboard ATS buffers. The commands are tagged with absolute execution times. The command is executed when the onboard clock matches its time tag.

2.4.5.1 MPS Organization

The MPS system organization and interfaces are shown in Figure 2.4. The MPS functionality can be described in terms of its major subsystems.

1. Activity Plan - provides the functionality to create, view and modify the activity plan. The current TRACE MPS software manages the activity plan as a number of plan periods which each cover a unique load time period. Each plan period may have multiple command loads, with one being designated as the prime load. The other loads are alternates for contingency use. For Swift, this subsystem was modified to manage activity plan periods that may overlap in time. This capability is needed to meet the requirement to load an ATS with up to seven days of commanding and to do this daily if necessary. The currently loaded ATS may need to be updated more frequently due to new GRB discoveries. Like Swift, the GLAST mission does not have the ability to perform patch loads.

This subsystem builds the binary form of the ATS command loads. The ATS loads are generated by expanding all referenced activities into their constituent commands and by converting all of the commands in the activity plan period into their binary format as defined in the ITOS command database. Reports for the plan periods, and loads are generated within this subsystem.

2. Activity Dictionary – is responsible for managing the activity definitions. The activity dictionary is the repository for all user-defined sets of command sequences. This subsystem allows the user to create, edit, and delete activity definitions in the dictionary. Each activity definition has a name that can be used to reference the activity from an ATS command load or from other defined activities. The execution sequence of each command and activity reference within an activity definition is indicated by a relative time tag. Activities are designated as ATS-capable or RTS-capable. ATS-capable activities may be defined to accept passed parameters that can be assigned to one or several command submnemonic values. Parameters allow the user to delay providing submnemonic values until the activity is actually used. RTS-capable activities cannot have parameters, since they are used to create RTS table loads.

3. Constants – manages the constants definitions. Constants can be defined for various purposes, such as defining the maximum time span of an activity plan, the maximum number of commands in the ATS tables, the size or the ATS tables, and flags for automatic archiving of reports. Constants can be modified through the GUI.

4. Constraints File – manages the constraints file. The constraints file provides a means for the FOT to define command constraints for sequence dependencies, timing dependencies and time delay constraints. Currently, the GLAST mission does not have any defined constraints.

5. Event File – currently manages the events derived from the orbit products and contact schedules.

> *Modification Note:* The TRACE MPS software was modified for Swift to also manage preplanned science timeline events. This capability will not be needed by GLAST. The GLAST science timeline contains time-tagged spacecraft or instrument commands as do other missions using MPS.

The five types of orbital and contact events are: pass events, duration events, ascending and descending node events, apogee and perigee events, and extended precision vector load events. Some events are instantaneous events and have only one time, while other events are have a duration indicated by a start time and a stop time. Events also include parameters that further characterize the event. For example, a pass event includes station ID, data rate, dump, ranging, tracking mode, point of closest approach, maximum elevation, and beta angle.

Events may be selected from the input files by time range and/or filtered by type of event and event parameters. Once selected, events and event parameters may also be modified by the user. New events may also be manually added to the event file.

6. Main Window – provides a framework for user interface event handling. This subsystem handles the initialization and event-handling functions that are common to all X Window System based applications. It instantiates all of the other top-level system objects and facilitates inter-object communications.

7. Project Database – manages the command definitions received from ITOS. It reads and validates the raw database files, provides access to command template definitions and defines methods for translating commands from text to upload format. Only commands defined in the database can be entered into triggers, activities, and activity plans. When updated raw database files are installed, the MPS produces a report that indicates any activities, trigger definitions, constraints, or loads that are affected by the changes in the command definitions.

8. Science Manager - manages the science files. It allows the user to select and deselect the set of science timeline files to use during the triggering process. It also lets the user edit, delete, and print the science files. *GLAST will use this feature to ingest science command files from the GSSC.*

9. Spacecraft Model – models the contents of the stored command buffers onboard the spacecraft. The ATS model tracks the contents of ATS A and B buffers. The RTS model tracks the status of each RTS table by managing the past, current, and future configurations of each table.

10. RTS Manager – manages the set of RTS table loads that have been created by the MPS. Using the RTS manager, the user may create, delete, and print RTS loads; view information

associated with previously created RTS loads, and record the uplink time of uploaded RTS loads.

11. Trigger Dictionary – manages the trigger dictionary. It allows the user to create, edit and delete triggers as well as save, revert and print the contents of the trigger dictionary. Event triggers are manually created by the FOT to automatically schedule commands and activities based on the contents of the event file. Each trigger defines a set of conditions. When these conditions are met by one or more events the trigger is said to fire. When a trigger fires, it causes the associated set of commands and activities to be placed in the activity plan. Absolute time tags are assigned to these commands and activities based on either the start or stop time of the event that caused the trigger to fire.

12. Report Generation - a common set of software that handles report formatting and report archiving. All subsystems are responsible for generating and archiving their respective reports using the report generation utilities. The reports that can be generated are:
- Integrated Observatory Timeline Report – all events and uplinks for a period
- Detailed ATS Command List Report (Stored CMD Log) – time-tags and commands
- Current Spacecraft Configuration Report – memory buffer usage
- Pass Plan Report – station contacts and uplinks
- Command Database Installation Report – results of installing the PDB
- Event File Report – all events in the event file
- Activity Dictionary Report – all activity definitions in the dictionary
- Trigger Dictionary Report – all trigger definitions in the dictionary
- Science Timeline Report – validation report
- ATS Load Report – result of ATS load generation; errors and warnings
- RTS Load Report - result of RTS load generation; errors and warnings
- RTS Memory Map Report – RTS table assignments
- Event Log Report – event log messages for a time period

2.4.5.2 Typical Processing Flow

Referring again to Figure 2.4, the typical ATS load planning session begins when the FOT uses the MPS Event/File Handler to integrate the orbital events, contact schedules, the science timelines and manual FOT event inputs to create a pool of events. The FOT then initiates the trigger process which uses the event pool and trigger dictionary to determine if any trigger definitions are active (fired). The MPS stores the triggered commands and activities in the activity plan.

The FOT selects an activity plan period from which to create an Absolute Time Sequence (ATS) command load. The duration of an activity plan period can be adjusted by the FOT. Then the FOT may edit the command load definition to include additional commands and activities. The command load definition may contain individual commands, references to ATS-capable activities, and references to RTS-capable activities.

The FOT initiates load generation to create the ATS command load file. The load generation process replaces each reference to an RTS-capable activity with the RTS start command and uses the spacecraft model to determine the table assignment. It uses the activity dictionary to expand the ATS-capable activities. If any command constraints are defined, the command list is checked against the constraints. Then the command list is converted into a binary load file using the definitions from the command database. The successfully generated load can then be transferred to the MOC server. This process is illustrated in Figure 2.4.5.2.

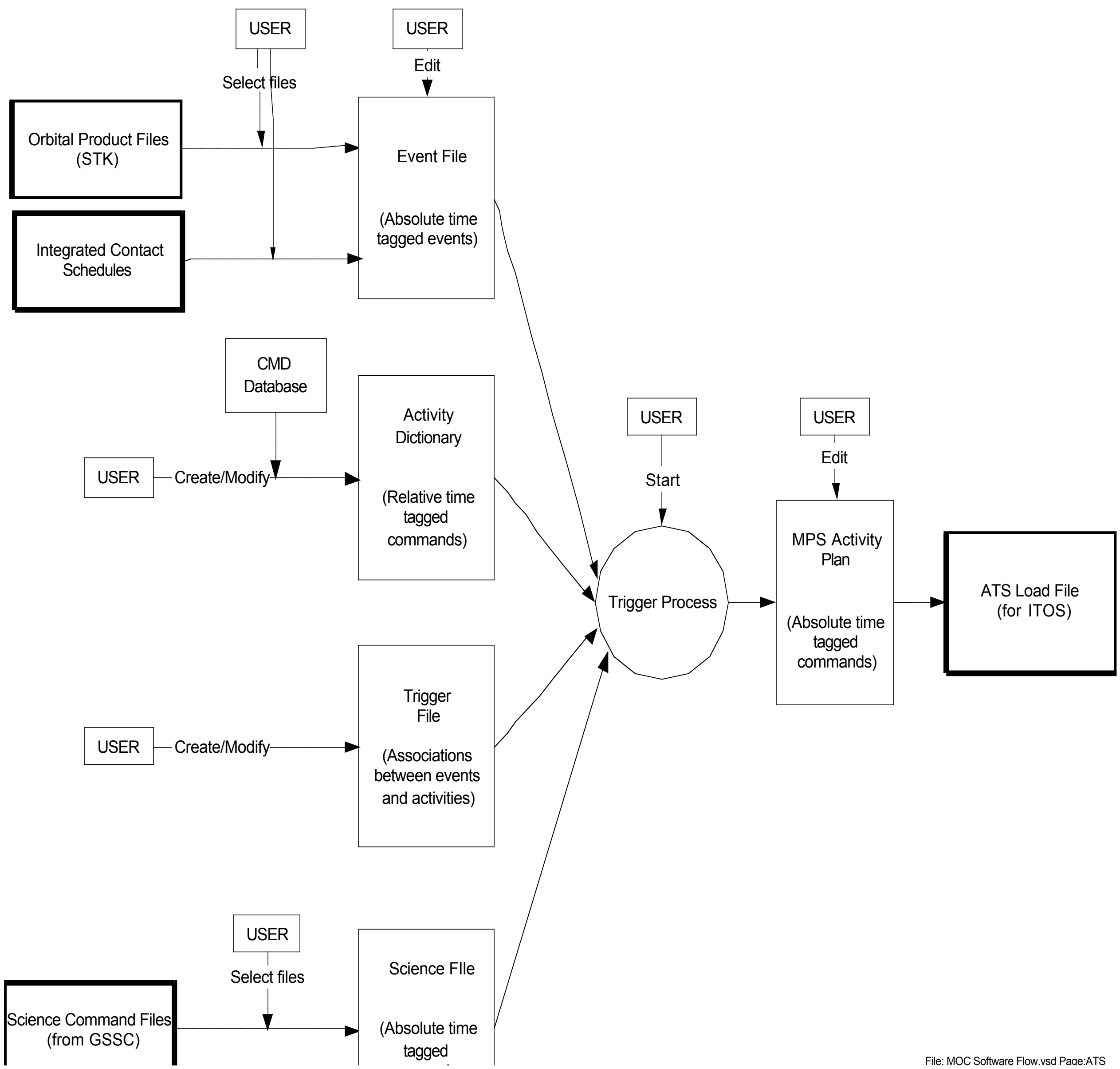


Figure 2.4.5.2 – GLAST MPS ATS Creation Process Diagram

The RTS table loads are generated for command sequences that are to be stored in the onboard RTS tables. RTS tables typically contain command sequences are executed routinely as well as command sequences needed for contingency events, such as safe-hold. The RTS loads must be created before attempting to generate an ATS load that references the RTS-capable activity.

Using RTS manager, the FOT selects an activity from a list of RTS-capable activities. The FOT selects an available table destination for the load and sets the uplink time range and the expiration date. The spacecraft model keeps track of which tables are available and the command database provides the binary command information. After the RTS load is generated successfully, the FOT transfers it to the MOC server.

2.4.5.3 Feature summary

A summary of MPS features that fulfill the GLAST MOC requirements are:

- Provide functionality to create, view and modify the activity plan.
- Ability to create multiple command loads for the same or overlapping time periods.
- Generate the binary form of the ATS and RTS command loads.
- Generate reports for plan periods, load or activities.
- Ability to create, edit, save and delete activities.
- Detect changes needed to loads and activities due to command database updates.
- Ability to define constants to customize MPS functions:
- Maximum time span of activity plan
- Maximum number of command in ATS table A and B
- Size of ATS table A and B
- List of RTS tables available for MPS allocation
- Flags for automatic archiving of reports.
- Ability for the FOT/SOT to define command constraints for:
- Sequence dependencies
- Timing dependencies
- Time delay constraints
- Read and validate new raw project database files
- Ability to select/deselect the set of science timelines files to use during triggering process.
- Ability to edit, delete, and print the science timeline files.
- Maintain the configuration of the ATS A and B buffers.
- Maintain the past, current, and future configurations of each RTS table onboard the spacecraft.

- Ability to create, edit, and delete triggers as well as save, revert, and print the contents of the trigger dictionary.

Design Notes:
Suggested enhancements for GLAST (and future ITOS-based missions)
1. Uncomment the science command file input feature. It was hidden in the Swift version
2. Ingest ITOS-format database directly instead of PDB-format. Info is lost in the translation
3. Upgrade MPS command database for more data types and kinds of submnemonics
4. For testing – add a time offset from realtime for I&T and launch tests – needed to test "plan purge" and event ingest time limits
5. Investigate adding GMSEC compliant interface.

2.4.6 Attitude-dependent TDRSS Scheduling

With GLAST, the scheduling of TDRSS contacts is complicated by the location of the Ku-band antenna on the spacecraft. At times during a planned observation, GLAST may be pointed so that the antenna cannot transmit to any TDRS. Therefore the MOC must analyze the science timeline to determine when TDRSS contacts are physically possible.

The Attitude-dependent TDRSS Scheduling software component uses the planned observation timeline created by Science Input Processor (*plus ToOs? TBD*) to create a predicted attitude file. Using STK/Connect, this file is input to STK/Pro which creates a predicted TDRSS view report. The FOT selects and schedules 4 or 5 contacts from this report.

The Attitude-dependent TDRSS Scheduling software performs these processing steps:

1. Create the predicted attitude file

Use the planned observation timeline to create a predicted attitude file.
Also, the attitude file will have to model s/c motion times since we can't have a Ku-band contact during a slew. Slews will have to be modeled for:
1. ToO/PO entrance and exits.
2. Yaw flips
3. Rocking sky survey mode transitions (when using a stepped profile.)
4. Rocking mode sky survey mode (when using a sinusoidal profile, if one exists).
5. Earth limb trace (when primary target is occulted by the earth.)

Design Issues:
1. What format does STK need? coordinate type? time resolution?
2. How would ToOs be included?

2. Create TDRSS view reports

Uses STK/Connect to configure and initiate STK/Pro. Based on the predicted attitude file, spacecraft antenna mask defined in STK, the predicted orbit ephemeris for GLAST and TDRSS, STK produces GLAST to TDRSS view reports.

3. Schedule TDRSS Contacts

After the TDRSS view reports are created, the software brings up an editor panel that lists the accesses. The user can set filters for pass length, preferred TDRSS, slewing times, etc. STK scheduler or MOC script attempts to choose the best passes based on preset criteria i.e. length between passes. The MOC analyst reviews the schedule for number of passes and length between them and edits where necessary. MOC schedules the 4 to 5 passes per day with SWSI.

> *Design Issues:*
> *Scheduling could be done automatically if there is an automated interface with SWSI or with replacement system (SNAS).*

2.5 Trending and Analysis

The Trending and Analysis subsystem consists of three software entities, DTAS, a web based DTAS client and a local DTAS client. These entities are depicted below in Figure 2.5 Analysis and Trending Software Flow Diagram, along with their external and internal interfaces.

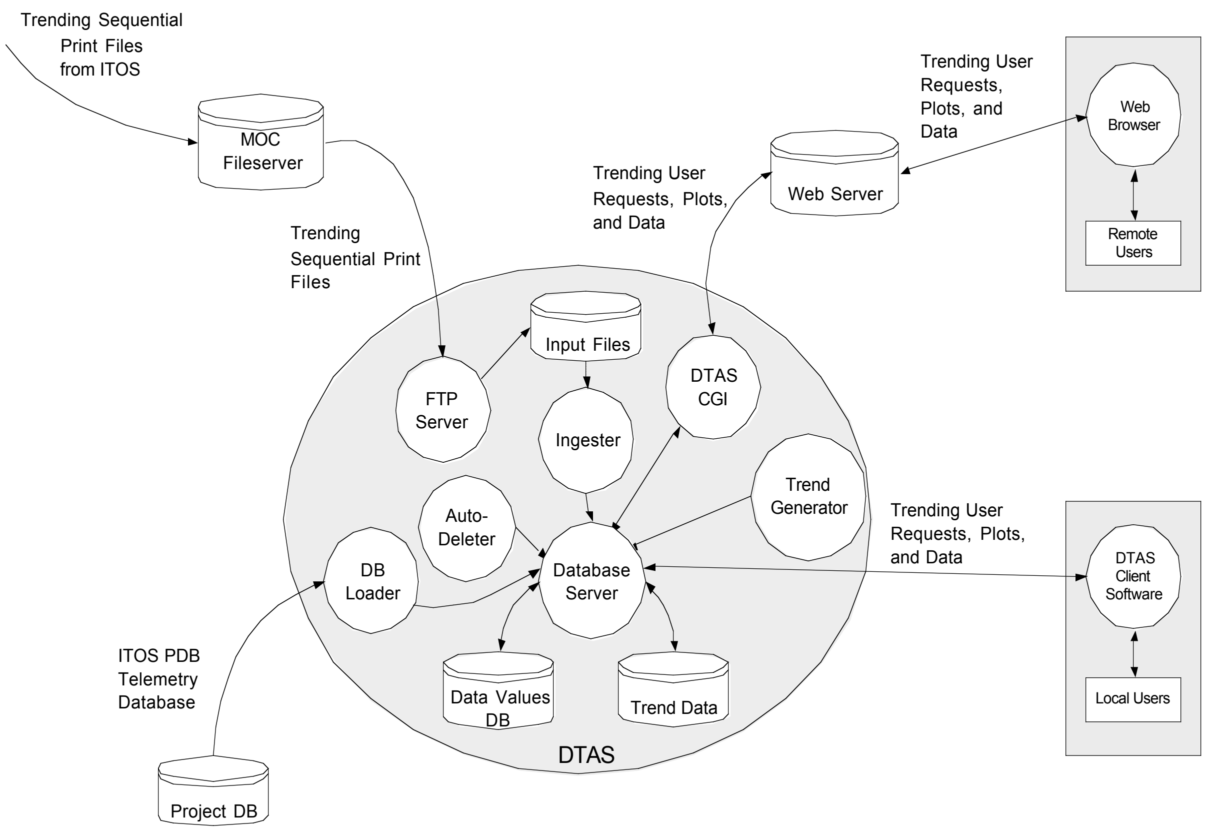


Figure 2.5 – Trending and Analysis Software Flow Diagram

2.5.1 Data Trending and Analysis System (DTAS)

The DTAS is a standalone GOTS software package that provides telemetry data trending and analysis processing capabilities:

- Automated ingest of telemetry data and storage of data subsets
- Generation of graphical and numeric plots and reports of historical housekeeping data
- Statistical analysis
- Data extraction and filtering
- Export data to MS Excel for further analysis
- View, print, and save reports to a file
- Remote access via Internet

The DTAS subsystem consists of the following components:

<u>FTP Server</u>
The DTAS ingests the ITOS sequential print (SEQPRT) files automatically from the MOC file server when detected in the DTAS input directory. An FTP server performs this transfer

placing the SEQPRT files in the Input Files data store. Optionally a user can manually ingest selected data.

Input Files Data Store
Location in which the FTP Server stores SEQPRT files when received from the MOC File Server.

Ingester
The Ingester process reads the input SEQPRT files and stores the data in the Data Values DB via the Database Server.

> Note:
> The DTAS Ingester polls the input ftp directory for the presence of a new file (e.g. every 10 minutes), and may start on a partial file that can cause a processing failure. Manual intervention is required to re-ingest the data and process.

Database Server
The Database Server is a COTS product (Interbase) and is responsible for managing the storage and retrieval of the telemetry and trend data.

Auto Deleter
DTAS operates on a sliding window (e.g. 30 days) so the old data is automatically deleted from the Data Values database. This activity is performed by the Auto Deleter utility.

DB Loader
The DB Loader extracts the relevant telemetry parameters from ASCII files in the Project DB and loads the DTAS DB.

Data Values DB
This database contains the time and value for each mnemonic ingested.

DTAS CGI
An Apache Web Server and a DTAS Common Gateway Interface (CGI) script serve user requests for data

Trend Generator
The DTAS Trend Generator filters the Data Values DB to automatically generate trend data including min, max and mean at a selected sampling interval (e.g. 1 per orbit).

Trend Data
This data set contains subsets of housekeeping data sampled at selected intervals that are accumulated over the life of the mission.

DTAS Clients

DTAS provides the capability for the user/client to view plots and tables of the housekeeping and trend data and save these products to a file. The user can also request data extraction to a file or Excel spreadsheet. Data requests can include filter criteria such as time range and selected mnemonic values meeting a logical expression. A client request for data from the server is accomplished by SQL queries that can include filtering criteria.

DTAS has two types of clients, local and web based:

Local DTAS Client
With the local DTAS client the application software is installed on a user workstation using an install shield which can be downloaded via the MOC web site (e.g. remote users at the S/C Vendor or Instrument Team facilities). The client application consists of a Trending Tool and an Analysis Tool. The Trending Tool allows a user to plot individual mnemonics vs. time including min, max and mean. The Analysis Tool allows the user to view tables, plots, and statistics, and to compare data.

Web Based DTAS Client
The DTAS web client utilizes a web browser to access the server via a CGI script running on the web server. The functionality using this interface is limited to the ability to request a plot of a single mnemonic over a time range.

Design Issue:
Candidate for replacement with GMSEC-compliant trending and analysis system. The replacement would communicate with ITOS via the GMSEC architecture instead of via sequential print files.

2.6 MOC Web and Remote Access

The MOC Web and Remote Access subsystem consists of two software entities, ITOS and the Web Server. The Web Content Dispatcher does not reside on the same machine or network as the ITOS or web server but is shown because it is the application program interface (API) which the MOC applications use to push information to the web server. These entities are depicted below in Figure 2.6 - MOC Web and Remote Access Software Flow Diagram, along with their external and internal interfaces.

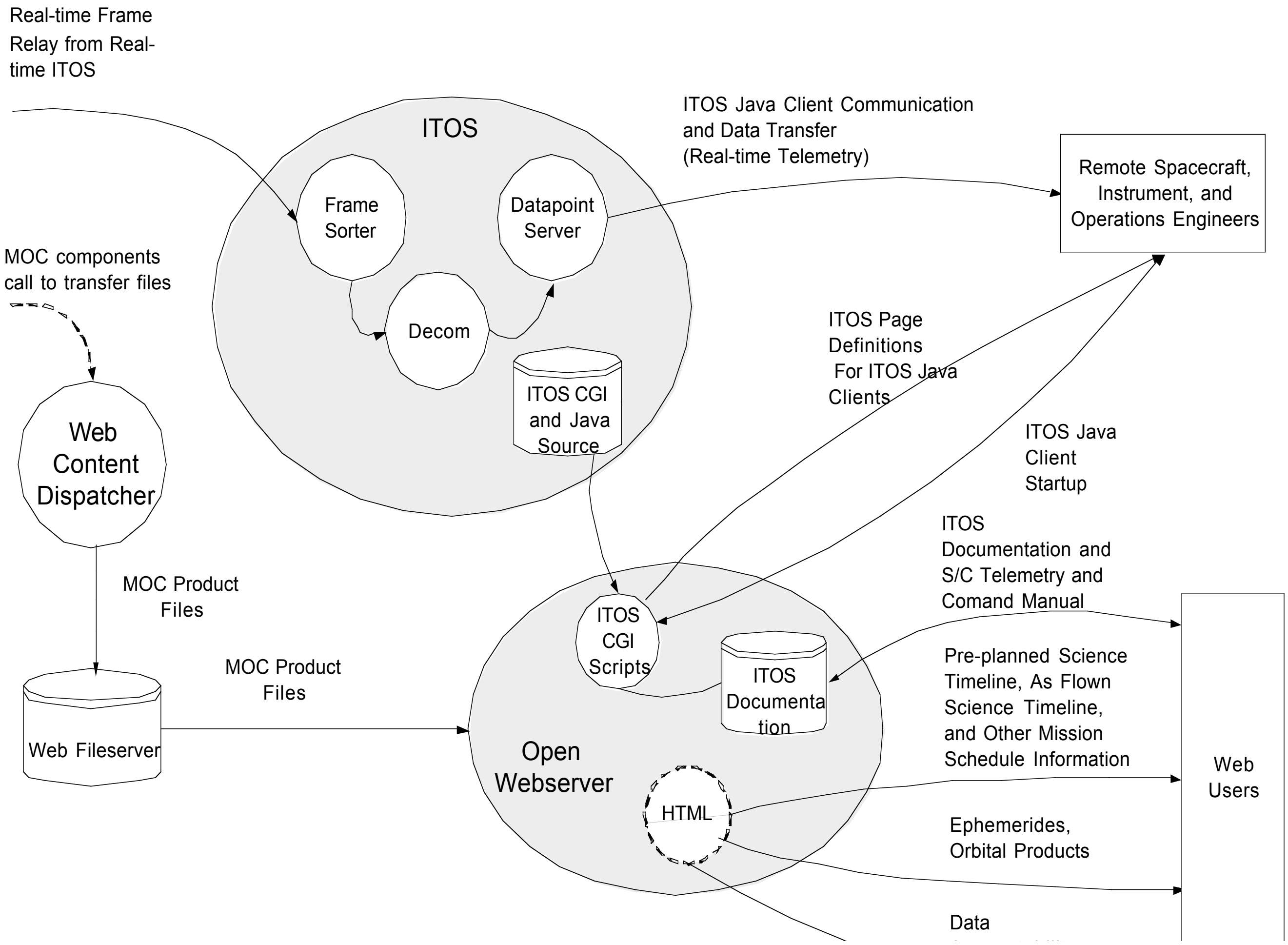


Figure 2.6 – MOC Web and Remote Access Software Flow Diagram

2.6.1 Integrated Test and Operations System (ITOS)

The ITOS subsystem executing on the open machine (i.e. outside the MOC closed subnet) will be responsible for receiving and processing real-time telemetry frames from the ITOS subsystem executing on the real-time telemetry machine located inside the firewall. The ITOS system will perform frame sorting, decommutation in order to populate the current value table located in the ITOS database. Remote Spacecraft, Instrument and Operations engineers will launch the ITOS Java client via a web browser. The Java client will make a socket connection to the ITOS Datapoint Server which will stream data from the current value table in the ITOS database to the Java client for real-time display.

See Section 2.2.1 for additional information regarding the ITOS.

2.6.2 Open Web Server

The Open Web Server executes on the open machine (i.e. outside the MOC closed subnet) just as the ITOS subsystem. It has two primary functions: to service the instantiation of an ITOS Java client initiated by a remote spacecraft/instrument/operations engineer, and to service requests made by users of the flight operations web page.

2.6.2.1 ITOS Java Client Script

The ITOS Java client script is responsible for launching the ITOS Java client when initiated to do so by a remote spacecraft/instrument/operations engineer. Once the ITOS Java client is launched it makes a socket connection to the ITOS Datapoint server. The ITOS Java client script is GOTS software and merely needs to be installed on the target platform in the proper directory.

2.6.2.2 MOC Web Page

The MOC web page is a web interface that will be used for viewing ITOS documentation, the GLAST Telemetry and Command Manual, and MOC data products such as the Preplanned Science Timeline, the As-Flown Timeline, NORAD TLEs and Data Accountability Information. A web user will use a browser to point to the URL of the password-protected MOC web page. Upon successfully entering the correct user name and password the user will be presented with a page that lists the MOC categories and products available for viewing or downloading.

For more detailed information regarding the MOC web page, see the “MOC Web Site – Detailed Design/Programmers Guide”.

2.6.2.3 Web Content Dispatcher

The Web Content Dispatcher is the mechanism for pushing MOC web content to the MOC web server. It provides the interface between the MOC applications and the MOC web server. It is invoked by a MOC application whenever that application needs to push one if its products to the web server for display.

For more detailed information regarding the Web Content Dispatcher, see the “Web Content Dispatcher – Detailed Design/Programmers Guide”.

3.0 MOC Operational Data Flow

This section describes the three main high level processing threads that show the data flow between the software components and the external elements.

3.1 Real-time Command, Telemetry and Product Data Flow

The data flow diagram in Figure 3.1 – Real-time Command, Telemetry and Product Data Flow, describes the MOC real-time processing thread. During a contact session with the TDRSS or ground station, the prime ITOS systems receive transfer frames and station status from the TDRSS (or connected ground station) and process and distribute resulting data and products to other systems internal and external to the MOC. The prime ITOS also performs any commanding activities with TDRSS or the ground stations. The VMOC/SERS systems receive and process ITOS event logs and page personnel in the event of an anomaly. Real-time operations can be performed in automated or staffed modes.

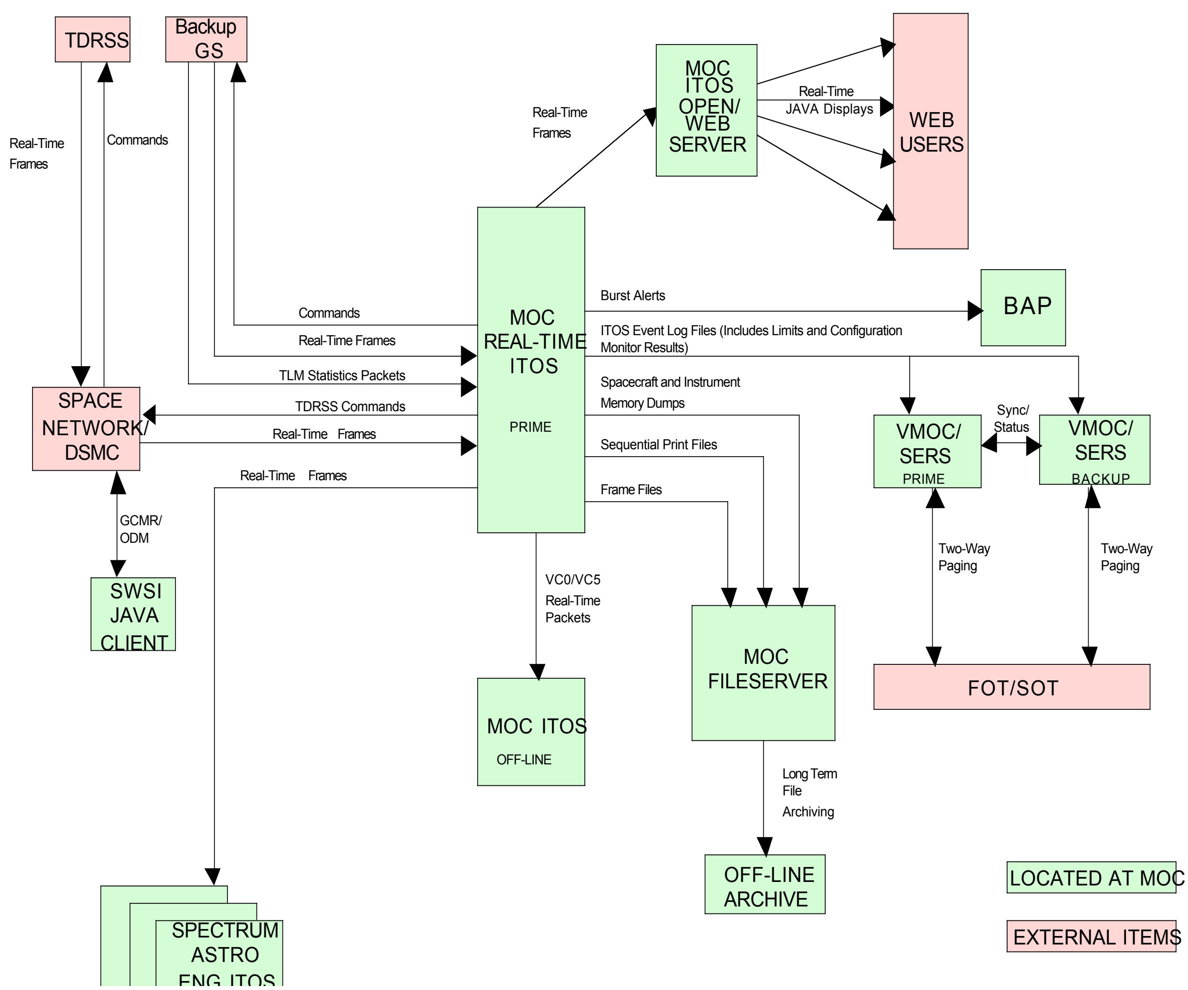


Figure 3.1 – Real-time Command, Telemetry and Product Data Flow

3.2 Offline Telemetry Data Flow

The data flow diagram in Figure 3.2 – Offline Telemetry Data Flow describes the MOC offline processing thread. The primary offline flow starts as the MOC receives spacecraft transfer frame files from the TDRSS after each TDRSS contact. The MOC ITOS offline system then processes the transfer frame files and distributes the resulting Level 0 datasets, telemetry data, and other products to other systems internal and external to the MOC. The VMOC/SERS systems receive and process ITOS event logs and page personnel in the event of an anomaly. Most of the offline software flow works in an automated flow requiring no operator interaction in the nominal situation.

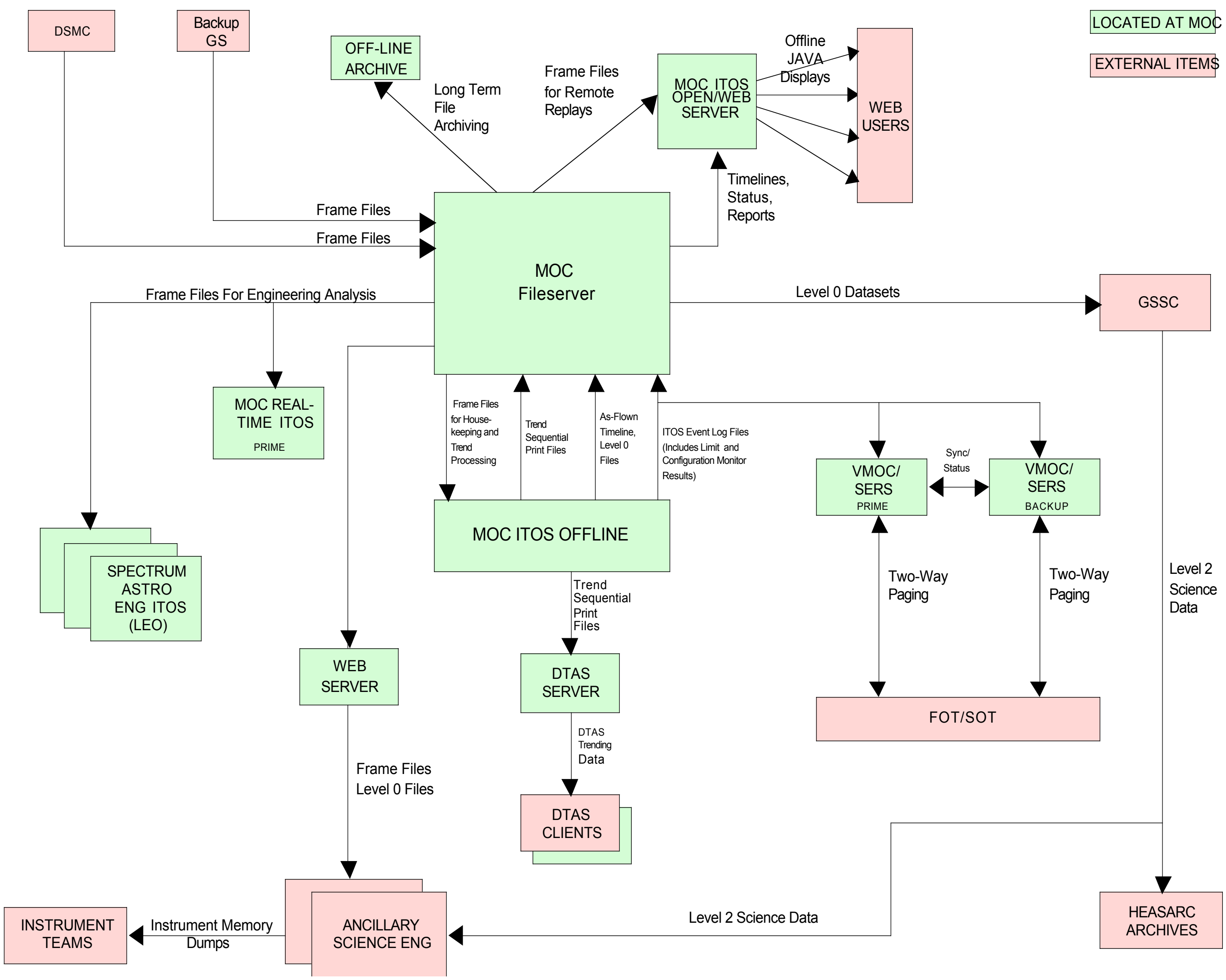


Figure 3.2 – Offline Telemetry Data Flow

3.3 Mission Planning Data Flow

The data flow diagram in Figure 3.3 – Mission Planning Data Flow describes the MOC mission planning processing thread. GLAST orbit determination is accomplished via Satellite Tool Kit, using GPS data from telemetry or, as a backup, publicly available NORAD TLE sets. Science activities are scheduled by the GSSC. The GLAST Mission Planning System (MPS) assembles ATS and RTS load files according to the supplied orbital products from STK, planned science activities from the GSSC, and integrated TDRSS/ground station contact schedule. The ATS and RTS loads are supplied to ITOS for transmission to the spacecraft.

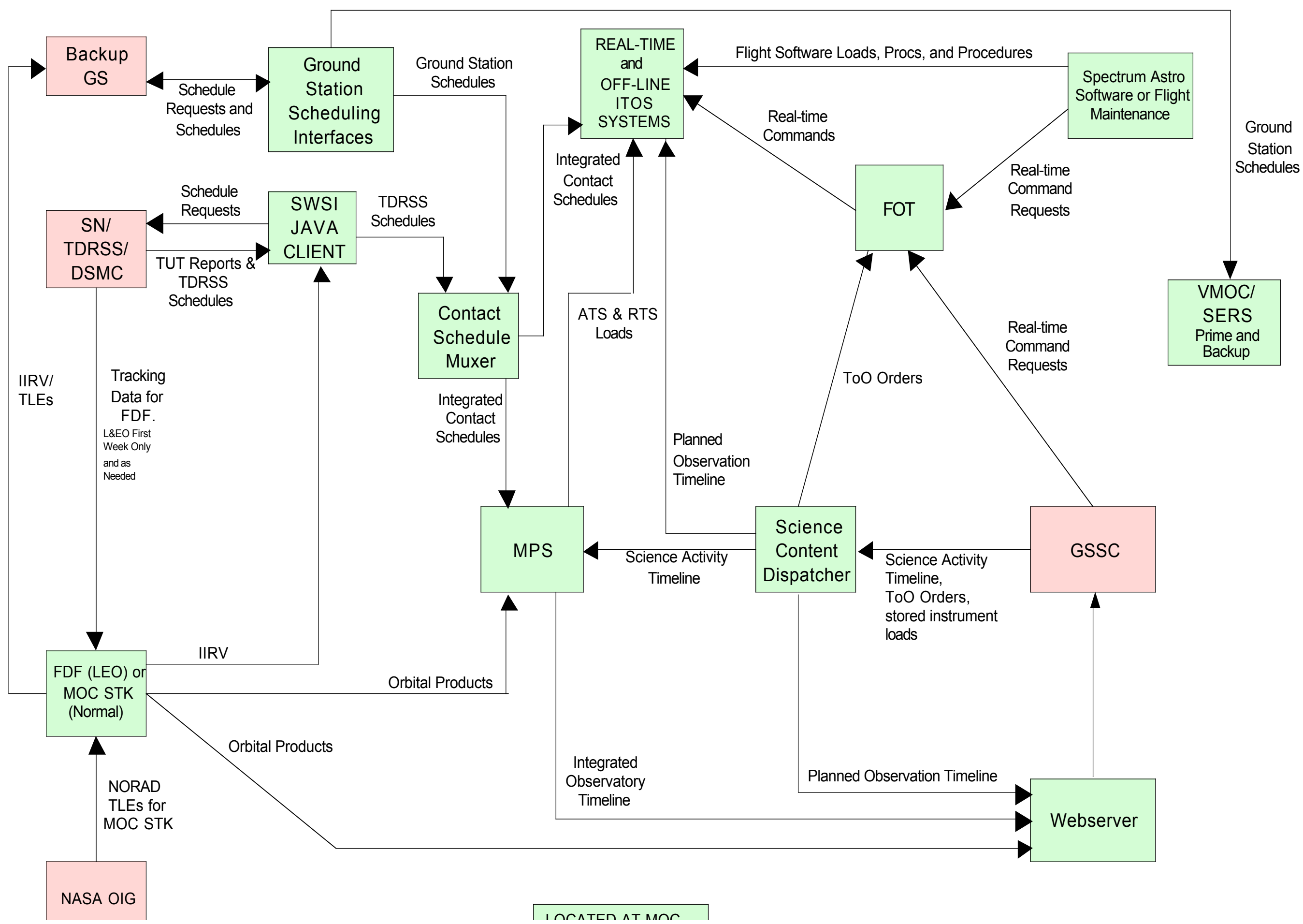


Figure 3.3 – Mission Planning Data Flow

3.4 System Start-up and Processing Timeline

The following tables describe the system startup and processing timelines for the Real-time Telemetry and Command Subsystem, the Mission Monitoring and Offline Processing Subsystem, and the Mission Planning & Scheduling Subsystem.

Table 3.4-1 - System startup and processing timeline for the Real-time Telemetry and Command Subsystem.

Event Time	Event Type	Event
	Terminal	Realtime AMAC is manually started
	AMAC	Reads integrated contact schedule and create contact task schedule
	AMAC	Read the contact task schedule (current or edited)
Immediate	AMAC	Starts ITOS if not already started
AOS – 15	AMAC	Starts RT AMAC Contact Proc
AOS – 15	PROC	Display the page showing current, past and future contact status
AOS – 14	PROC	Initialize ITOS telemetry and command subsystem
AOS – 14	PROC	Open telemetry, command, and station status packet server sockets
AOS – 5	PROC	Open VC0 frame archive.
AOS – 5	PROC	Start VC0 frame relay to open server and other external systems if necessary.
AOS – 5	PROC	Start SSR read pointer sequential prints.
AOS	PROC	Start configuration monitor checking at receipt of data.
AOS + Throughout	PROC	Limit violation checking
AOS + Throughout	AMAC	Process ITOS event logs
AOS + 1	PROC	When receiver and CDU lock is verified, send NOOP command to spacecraft.
AOS - LOS	STOL	If pass in manned, perform any manual operations necessary. Manual operations would primarily consist of interactive ITOS STOL PROC running, commanding, and telemetry verification.
LOS + 1	PROC	Close VC0 frame archive
LOS + 3	AMAC	Capture station frame statistics and spacecraft SSR dumping statistics and indicate in ITOS event log if there is a large discrepancy.
LOS + 3	AMAC	Indicate through event log of general pass status.
LOS + 5	AMAC	Transfer VC0 frame archive to MOC file server
LOS + 5	AMAC	Distribute SSR read pointer sequential print files to the MOC file server
LOS + 5	AMAC	Initiate event delogger program which will parse and distribute event log to various systems.
LOS + 7	AMAC	Updates contact status in the contact task schedule
LOS + 7	AMAC	Wait for next AOS and repeat above timeline

Table 3.4-2 - System startup and processing timeline for the DAS Telemetry Subsystem

Event Time	Event Type	Event
	Terminal	DAS AMAC is manually started
	AMAC	Reads integrated contact schedule
Continuous	AMAC	Reads the SWSI Alert Logs
Immediate	AMAC	Starts ITOS if not already started
Immediate	AMAC	Starts DAS AMAC Startup ITOS Proc
Immediate	PROC	Display the page showing TDRSS status
Immediate	PROC	Initialize ITOS telemetry subsystems
Immediate	PROC	Open TDRSS server sockets
Immediate	PROC	Open VC5 frame archive
Immediate	PROC	Start VC5 frame relay to open server and other external systems if necessary
Immediate	PROC	Start GRB sequential prints
Immediate	PROC	Start configuration monitor checking
Immediate	PROC	Start limit violation checking
GS AOS and every 15 minutes	AMAC	Start DAS AMAC Management Proc
GS AOS and every 15 minutes	PROC	If started at the beginning of a scheduled TDRSS or GS contact, closes old VC5 frame archive if there is data in it and opens new VC5 frame archive. Then sends old VC5 frame archive file to MOC archive for Level 0 processing.
GS AOS and every 15 minutes	PROC	Closes old ITOS event log and opens new ITOS event log. Initiate event delogger program which will parse and distribute event log to various systems.
GS AOS and every 15 minutes	AMAC	Generate a report of the last TDRSS contact statistics
Throughout	AMAC	Process ITOS event logs
	AMAC	Wait for AOS or 15 minute timer and repeat activities starting at DAS AMAC Management Proc

Table 3.4-3 - System startup and processing timeline for the Mission Monitoring and Offline Processing Subsystem.

Event Time	Event Type	Event
	Terminal	Offline AMAC is manually started
Continuous	AMAC	Poll the file server for telemetry files If no telemetry files are found, continue to poll If telemetry files are found, start up an instance of ITOS
Receipt of new telemetry files	AMAC	• Start the Frame Accounting Software • Distributes Frame Accounting results to fileserver • Start the Offline AMAC ITOS Processing Proc
Receipt of new telemetry files	Proc	Start telemetry file playback Start CCSDS packet extraction to Level 0 Datasets Start DTAS sequential print file data capture on VC1 and VC6 playbacks Start ITOS sequential print data capture of Attitude Information message (SISCATTITUDE for VC6 only) Start ITOS sequential print data capture of Figure of Merit (FOM) to instruments Next Observation Info Start Message (FONEXTOBSINFO for VC 6 only). Start ITOS Event Log Capture Start Configuration monitors Indicate through event log that VC1 and VC6 processing has completed Distribution of DTAS sequential print files to DTAS and trending archive. Distribution of SISCATTITUDE and FONEXTOBSINFO sequential prints to MOC archive (VC6 only). Distribution of Quick-look Level 0 Datasets to MOC file server and the GSSC. Initiate event delogger program which will parse and distribute event log to various systems.
Throughout telemetry file playback	ITOS	Configuration monitor and limit violation checks
Throughout telemetry file playback	AMAC	Process ITOS event logs
Completion of telemetry file playback	Proc	• Indicate through event log that processing has completed • Distribution of DTAS sequential print files to DTAS and trending archive. • Distribution of SISCATTITUDE and FONEXTOBSINFO sequential prints to MOC archive (VC6 only). • Distribution of Level 0 Datasets to MOC file server and the GSSC. • Initiate event delogger program which will parse and distribute event log to various systems.
Completion of telemetry file playback	AMAC	Start the Timeline Monitor if the telemetry file just processed was from VC6.
Completion of telemetry file playback	AMAC	Repeat above timeline starting with polling the fileserver for new files

Table 3.4-4 - System startup and processing timeline for the Mission Planning & Scheduling Subsystem.

Event Time	Event Type	Event
As needed	External	Receive Ground Station schedule
10:00z-16:00z	Science Input Processor	Receive Science Activity Timeline from GSSC
15:00z	Terminal	STK Automation Software is manually started
15:00z	STK Auto	Queries NASA OIG website for new TLE sets and retrieves them when one becomes available.
15:01z	STK Auto	Loads TLE set into STK and archives it on the Web Server and File Server
15:02z	STK Auto	Generates ground station and TDRSS view periods and orbit event files using STK/Pro and STK/Connect
15:03z	STK Auto	Converts ephemeris file to IIRV format and writes this file to the MOC file server.
15:15z-15:30z	SWSI	Manually download new TDRSS schedule using SWSI
15:30z	Terminal	Manually run Contact Schedule Muxer
16:00z-18:00z	Terminal	Manually create ATS and RTS loads using MPS system.

4.0 MOC FACILITY

This section describes the physical and logical architecture for the MOC. It provides an overview of the physical location of the MOC, the communication links, and voice system, as well as the hardware design and network layout. This section also describes the individual components that make up the MOC including a mapping of software to hardware.

4.1 MOC Network Configuration

The MOC provides a network of systems that are used by the Flight Operations Team (FOT) to perform spacecraft and instrument operations. Figure 4.1 provides an illustration of the MOC hardware and LAN architecture.

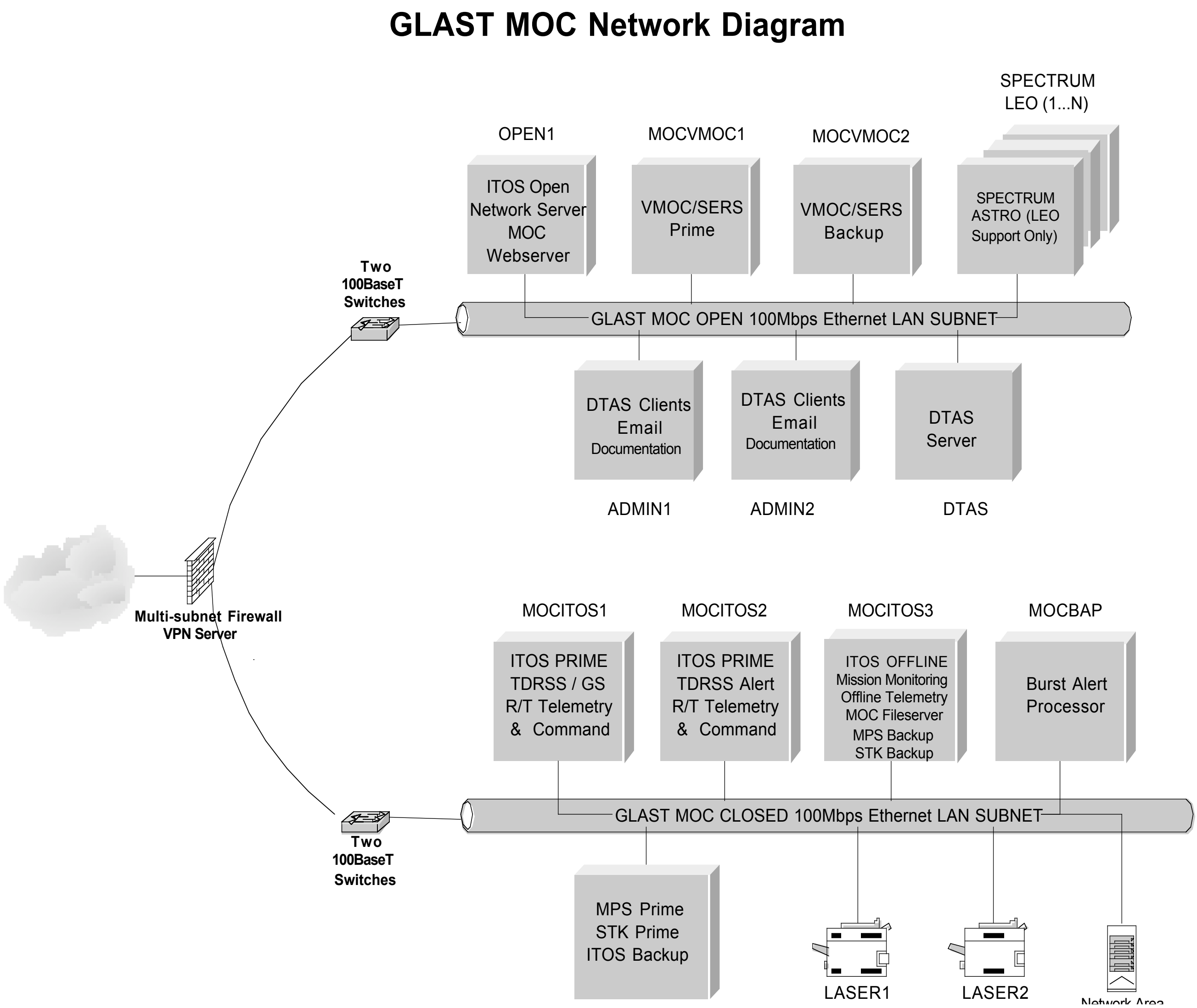


Figure 4.1 - MOC Hardware and Network Configuration

The MOC network is composed of two 100 Mbps Ethernet Local Area Network (LAN) subnets: MOC Open, and MOC Closed. For each subnet, the security and access is primarily controlled by a multi-subnet firewall which individually controls network access between the two subnets and the IONet and subsequently the Internet. Access to each subnet and computer is individually tailored at the firewall.

The MOC Closed subnet consists primarily of workstations running critical MOC operations: Real-time Telemetry and Command, Mission Monitoring and Offline Processing, and Missions Planning and Scheduling. Access from any external computers is severely restricted or not available.

The MOC Open subnet consists primarily of workstations running non-critical MOC operations: Web and Remote Access and Analysis and Trending. Access from external computers is permitted only for the specific services required.

Specific hardware and software information is available in the Hardware and Software Mapping Table in section 4.1.2.

4.1.1 Subnets, Firewall

TBS

4.1.2 Hardware Description

The MOC system consists of several kinds of workstations and peripherals. Each workstation is assigned to run certain components of the MOC software. A description of the function of each hardware component is given in Table 4.1.2.

Table 4.1.2 – GLAST MOC Hardware & Software Mapping Table

Workstation/ Hardware ID	MOC Hardware (subject to trades)	MOC Software	Function

Workstation/ Hardware ID	MOC Hardware (subject to trades)	MOC Software	Function
MOCITOS1 Primary Real-time Command & Telemetry Workstation	Sun Ultra 10/Solaris 8 21" Monitor 256MB Ram 20GB HD 48xCDROM	ITOS Realtime AMAC Event Delogger	Command & telemetry processing, STOL scripting, User Interface/Displays Automated control and monitoring for real-time telemetry from scheduled contacts (SN/GN) Automated ITOS event log monitoring and report generation
MOCITOS2 Primary DAS Command and Telemetry Workstation	Sun Ultra 10/Solaris 8 21" Monitor 256MB Ram 20GB HD 48xCDROM	ITOS DAS AMAC Event Delogger	Command & telemetry processing, STOL scripting, User Interface/Displays Automated control and monitoring for telemetry received via TDRSS DAS Automated ITOS event log monitoring and report generation
MOCITOS3 Mission Monitoring and Off-line Processing Workstation	Sun Ultra 60/Solaris 8 21" Monitor 256MB Ram 20GB HD 48xCDROM	ITOS Off-line AMAC Frame Accounting Software Timeline Monitor Data Archiver Event Delogger MPS (backup) STK (backup) STK Automation (backup)	Telemetry processing, STOL scripting, User Interface/Displays, Level 0 dataset processing Automated control and monitoring of all telemetry file processing Collection of missing frame and frame accountability statistics Monitor execution of observation timeline on spacecraft and creation of as-flown timelines Archive data file to tape Automated ITOS event log monitoring and report generation Backup MPS Backup STK Backup STK Automation

Workstation/ Hardware ID	MOC Hardware (subject to trades)	MOC Software	Function
MOCMPS Mission Planning Workstation	Sun Ultra 10/Solaris 8 21" Monitor 256MB Ram 20GB HD 48xCDROM	MPS STK STK Automation Contact Schedule Muxer ITOS (backup)	Creation and management of spacecraft ATS and RTS loads Generation of orbital products Automated control of the STK software Consolidate TDRSS and ground station contact schedules Backup ITOS
OPEN1 Open Workstation	Sun Ultra 10/Solaris 8 21" Monitor 256MB Ram 20GB HD 48xCDROM	ITOS MOC Web Site	Remote access to telemetry processing Provide access to MOC files: documentation, timelines, TLE, data accountability information
MOCVMOC1 Primary VMOC/SERS PC	Intel PC/Windows 2000 Pentium 4 17" Monitor 256MB Ram 40GB HD 48xCDROM	VMOC/SERS	Anomaly detection, issue pages to remote personnel
MOCVMOC2 Backup VMOC/SERS PC	Intel PC/Windows 2000 Pentium 4 17" Monitor 256MB Ram 40GB HD 48xCDROM	VMOC/SERS (backup)	Anomaly detection, issue pages to remote personnel
DTAS Trending and Analysis Workstation	Intel PC/Windows 2000 Pentium 4 17" Monitor 512MB Ram (2) 100GB HDs 48xCDROM	DTAS Server	Telemetry data trending and analysis processing
ADMIN1 Admin PC	Intel PC/Windows 2000 Pentium 4 19" Monitor 256MB Ram 40GB HD 48xCDROM	DTAS clients Email and office products	Local and remote plotting and reporting of trending data Administrative tools
ADMIN2 Admin PC	Intel PC/Windows 2000 Pentium 4 19" Monitor 256MB Ram 40GB HD 48xCDROM	DTAS clients Email and office products	Local and remote plotting and reporting of trending data Administrative tools
SPECTRUM (1 through N)	Intel PC /Solaris x86 8 19" Monitor 256MB Ram 20GB HD 48xCDROM	ITOS	Telemetry processing, User Interface/Display
Network Area Storage			MOC data storage server

Workstation/ Hardware ID	MOC Hardware (subject to trades)	MOC Software	Function
Offline Media			Large capacity backup device & backup software
100BaseT Switches (4)	3Com® SuperStack® 3 Switch 3300 XM Switch		Network switch
Multi-subnet Firewall/VPN Server	Intel PC/ Windows 2000 17" Monitor 512MB Ram 40GB HD 48xCDROM 5 Ethernet Cards	Checkpoint NG VPN-1	Firewall VPN remote access
LASER1	HP Laserjet 2200TN		Networked Laser Printer Prime
LASER2	HP Laserjet 2200TN		Networked Laser Printer Backup

4.2 Global Network Configuration

4.2.1 Communications Lines

The communication lines between the MOC and its external interfaces are depicted and defined below in Figure 4.2.1. – MOC Communications Lines, Figure 4.2.2 - GLAST WAN Architecture, and Table 4.2.1 – MOC External Interfaces Communications Lines, respectively.

TBS

Figure 4.2.1 – MOC Communications Lines

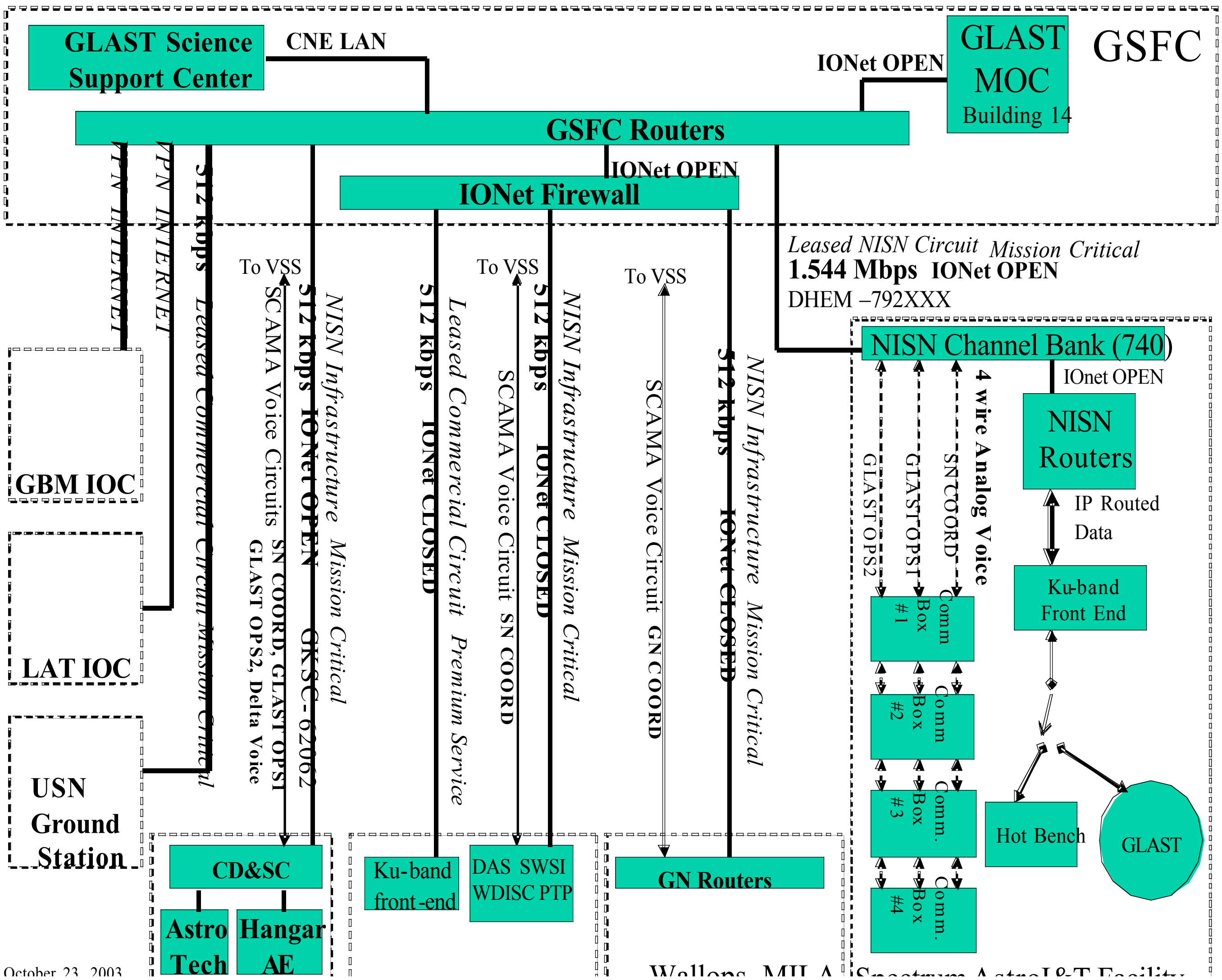


Figure 4.2.2 – GLAST WAN Architecture

Table 4.2.1 – MOC External Interfaces Communications Lines

Interface	Data Type (Rate)	Data Path Segments	Data Rate
MOC - Wallops	Real-time TLM (? kbps) Commands (? kbps) Playback TLM Files (ftp'd up to max line rate)	IONet	
		MOC◊IONet NISN Circuit	512 kbps
MOC - USN	Real-time TLM (? kbps) Commands (? kbps)	MOC◊USN Leased Circuit	512 kbps
	Playback TLM Files (ftp'd up to max line rate)	Internet	≥ 256 kbps
MOC – DAS/WDISC	Real-time TLM (? kbps) Commands (? bps)	IONet	
		MOC◊IONet NISN Circuit	512 kbps
MOC – Ku-band		IONet MOC◊Leased Circuit	512 kbps
MOC - GSSC	Science Input Products MOC Products	GSFC CNE	
		Internet	≥ 1 Mbps
MOC – GIOC	Science Input Products MOC Products	VPN Internet	
MOC – LIOC	Science Input Products MOC Products	VPN Internet	
MOC—SA I&T	Real-time TLM (? kbps) Commands (? bps)	IONet OPEN MOC -> Lease NISN Circuit	1.544 Mbps
MOC - FDF	Orbital Products	IONet	
		MOC◊IONet NISN Circuit	56 kbps

4.2.2 Router Configurations

TBS

4.2.3 Organization Responsibilities/Contacts

4.3 Voice System Configuration

The voice system consists of COTS components that provide the required conferencing and monitoring capabilities necessary for MOC operations. The voice system is composed of two major components:

1. Mission Ops voice (XteQ system) via IONet
2.

The voice network is depicted below in Figure 4.3 – GLAST Mission Voice System Network.

TBD

Figure 4.3 – GLAST Mission Voice System Network

4.3.1 Mission Ops Voice System

The Mission Ops voice system is composed of the PADS Series 600 Voice Terminal Units (VTUs) from XteQ, Inc. Each PADS VTU supports up to six, 4-wire analog circuits or voice-nets with accompanying headsets. The voice channel from the IONet router voice line termination point is connected to the first PADS VTU, and each subsequent VTU unit is connected in a daisy-chain fashion (see Figure 4.3.1). This daisy-chain arrangement allows any number of units to be added to the system as needed.

Reference the PADS VTU Product Specification for additional information.

[Figure 4.3.1 TBS]

4.4 MOC Software Configuration

The MOC software is delivered as a build that consists of a release of each software component. The installed files from older builds will be kept on the system in case there is a problem with the new build.

The COTS and GOTS software is installed locally on the target workstations listed in Table 4.1.2. The custom software for the MOC Closed subnet is installed on a shared disk drive (the MOC Fileserver) to make it easier to maintain. This arrangement also makes it easier to assign primary and backup workstations since they all have access to the code. The software on the MOC Closed subnet typically get input files from and put output files to the MOC Fileserver, which is described in the Section 4.4.1. Most of the applications, both custom and COTS, have configuration files which are data files that customize how the applications function for GLAST. The configuration files for the custom software are also stored on the MOC Fileserver under FS_MOC_CONFIG. The GOTS and GOTS software have unique locations for their configuration files. This is described in Section 4.4.3

The custom software for the MOC Open subnet will consist of web pages and scripts to support the MOC Web Site. The major data directory structure is for the web data.

4.4.1 MOC File Server

The MOC file server is simply a predefined directory structure residing on a SUN Solaris Ultra for the permanent and temporary archival of MOC system and product files. Since the MOC uses a file based mechanism for interprocess communication (IPC), the file server is a critical and essential system component within the MOC. The architecture of this directory structure is defined in the “mocenvrc” file. Any changes to the MOC file server directory structure should be made in this file. Once the mocenvrc file has been updated, the Perl script “make_fileserver_dirs.pl” should be executed so that the new MOC file server directories can be added.

The current MOC file server directories and intended contents are listed in Table 4.4.1. The MOC File Server Directory column lists the UNIX environment variable used by the custom software to refer to the directory. The use of environment variables allows software to be transparent to the physical directory path.

Table 4.4.1 – File Server Directories

MOC File Server Directory	Intended Content
FS_FTP	Top level FTP directory
FS_SERS1	Top level SERS1 directory
FS_SERS1_TDRSS	Event Delogger Failover directory for TDRSS Event Log
FS_SERS1_OFFLINE	Event Delogger Failover directory for Offline Event Log
FS_SERS2	Top level SERS2 directory
FS_SERS2_TDRSS	Event Delogger Failover directory for TDRSS Event Log
FS_SERS2_OFFLINE	Event Delogger Failover directory for Offline Event Log
FS_ITOS	Top level ITOS directory
FS_TLM_ARCHIVE	Top level ITOS telemetry archive directory
FS_LEVEL0	Level 0 products
FS_RT_TLM	Real-time telemetry files
FS_STATION_TLM	TLM frame files
FS_STATION_STATS	Station status packet files, Station telemetry statistics files
FS_ARCH_STN_STATS	Archive of TDRSS Station statistic files
FS_STN_SEG_STATS	TDRSS Segment Statistics files
FS_DAS_TLM	TLM frame files from DAS
FS_ITOS_INPUT	Top level ITOS input directory
FS_ITOS_ODB	ITOS operational database files
FS_ITOS_PDB	ITOS PDB format database files
FS_ITOS_CFGMON	ITOS configuration monitor definition files
FS_ITOS_LOADS	Top level loads directory
FS_ITOS_ATS_LOADS	Absolute Time Sequence Load files
FS_ITOS_FSW_LOADS	Flight Software Load files
FS_ITOS_RTS_LOADS	Relative Time Sequence Load files
FS_ITOS_PAGES	ITOS telemetry display page definition files
FS_ITOS_PROCS	ITOS STOL PROC files
FS_ITOS_OUTPUT	Top level ITOS output directory
FS_ITOS_DUMPS	ITOS memory and table dump files
FS_ITOS_PRINTS	Top level ITOS sequential print directory
FS_RT_SSR_SEQPRT	SSR sequential print output files
FS_READPTER_SEQPRT	SSR read pointer status sequential print output files
FS_TREND_SEQPRT	Trending sequential print output files
FS_ITOS_REPORTS	Top level Reports directory
FS_RPT_CFGMON	Configuration monitor events files
FS_RPT_CFG_DAS	Configuration monitor events files for DAS TDRSS
FS_RPT_CFG_OFFLINE	Configuration monitor events files for Offline
FS_RPT_CMD_ARCHIVE	Command events files
FS_RPT_CMD_RT_TDRSS	Command events files for DAS TDRSS
FS_RPT_CMD_OFFLINE	Command events files for Offline
FS_RPT_EVTLOG_ARCH	ITOS event log files
FS_RPT_EVTLOG_RT	Event Log files for RT
FS_RPT_EVTLOG_DAS	Event Log files for DAS TDRSS
FS_RPT_EVTLOG_OFFL	Event Log files for Offline
FS_RPT_LIM_ARCHIVE	Limit events files
FS_RPT_LIM_RT	Limit event files for RT
FS_RPT_LIM_DAS	Limit event files for DAS
FS_RPT_LIM_OFFLINE	Limit event files for Offline
FS_RPT_SWSI_ARCHIVE	SWSI events files
FS_PLANNING	Top level Planning directory
FS_IIRV	IIRV Files

MOC File Server Directory	Intended Content
`FS_DAS_IIRV`	DAS IIRV Files
`FS_LEGACY_IIRV`	Legacy IIRV Files
`FS_PPSTL`	Pre-planned science timeline files
`FS_AS_FLOWN_TL`	As-flown Timeline file
`FS_AS_FLOWN_TL_ARCH`	Archived As-flown Timeline file
`FS_AS_FLOWN_TL_WEB`	As-flown Timeline to be copied to the web
`FS_INTEG_OBS_TL`	Integrated observatory timeline files
`FS_PLAN_SCHEDULES`	Top level contact schedules directory
`FS_CURRENT_SCHED`	Latest contact/handover schedules
`FS_ICS`	Integrated contact schedule files
`FS_USN_SCHED`	USN contact schedule files
`FS_TDRSS_SCHED`	TDRSS contact schedule files
`FS_STK_PRODUCTS`	STK orbital products
`FS_PRED_SAA_PASSES`	Predicted SAA pass files
`FS_STK_SCENARIO`	Object files for STK
`FS_STORED_CMD_LOG`	Stored command log files
`FS_TLES`	NORAD TLE files
`FS_SOFT_CONFIG`	Software configuration files
`FS_GNDTASK_CONFIG`	Ground tasks configuration files
`FS_PAGING_CONFIG`	Paging criteria
`FS_MOC_CONFIG`	MOC Software Configuration files
`FS_TLM_PRODUCTS`	Telemetry products
`FS_CLOCKLOG`	Clock log files
`FS_TLM_FRAMESTATS`	Top level frame statistics directory
`FS_FRAMES_ACCT`	Frames accountability files
`FS_FRAMES_MISS`	Missing frames database
`FS_TLM_SIGNAL_FILES`	Level 0 signal files for GSSC
`FS_SIG_FILES_FO`	Failover directory for pass complete signal files

4.4.2 Installed Software Directories

This section describes the directory location of the MOC software on the operational workstations.

MOC Closed subnet

The COTS software components are installed in the following directory structure:
TBS

The custom software is installed on the MOC fileserver. All of the executable files for the current operational MOC configuration are available under `/filesvr/MOC_current/bin`. The files under the `bin` directory are links to the mostly recently installed build or in rare cases to some components from the previous build. The directory layout is shown in Figure 4.4.2-1 MOC Software Directory Layout.

TBD

Figure 4.4.2-1 - MOC Software Directory Layout.

4.4.3 Configuration Files

Some parameters for the MOC software may change over the various phases of the mission's life. To be responsive to changes and to reduce the number maintenance deliveries, many parameters of the custom software are set in a configuration file. Each custom application has its own configuration file. The configuration files can be changed by the user, so they are kept on the MOC file server, separately from the software executables.

The as-delivered version of the configuration file is available in the software installation directory (MOC_B*n*_install) under the application's conf directory. At installation, a writeable copy is placed in the $FS_MOC_CONFIG directory. The parameter values and usage are documented within the file and in the application's user guide. The custom application and the name of its configuration file are listed in Table 4.4.3.

Table 4.4.3 – Configuration Files

Application	Configuration Filename
Realtime AMACs	`RT_AMAC.conf` `TDRSS_AMAC.conf`
Contact Schedule Muxer	`contact_sched_muxer.conf`
Data Archiver	TBD
Frame Accounting Software	none
Mission Monitoring and Offline Processing Event Delogger	`evt_delogger.conf`
MPS	none
Offline AMAC	`offline_amac.conf`
Real-time Event Delogger	`evt_delogger.conf`
Science Input Processor	`sip.conf`
STK Automation	`stk_automation.conf`
Task Schedule Editor	`task_editor.conf`
Timeline Monitor	`as_flown_timeline.conf`
Web Content Dispatcher	`web_content_dispatcher.conf`

4.5 Facility Layout

4.5.1 MOC Building Description

The MOC facility is housed on the second floor of Building 14 at Goddard Space Flight Center, Maryland.

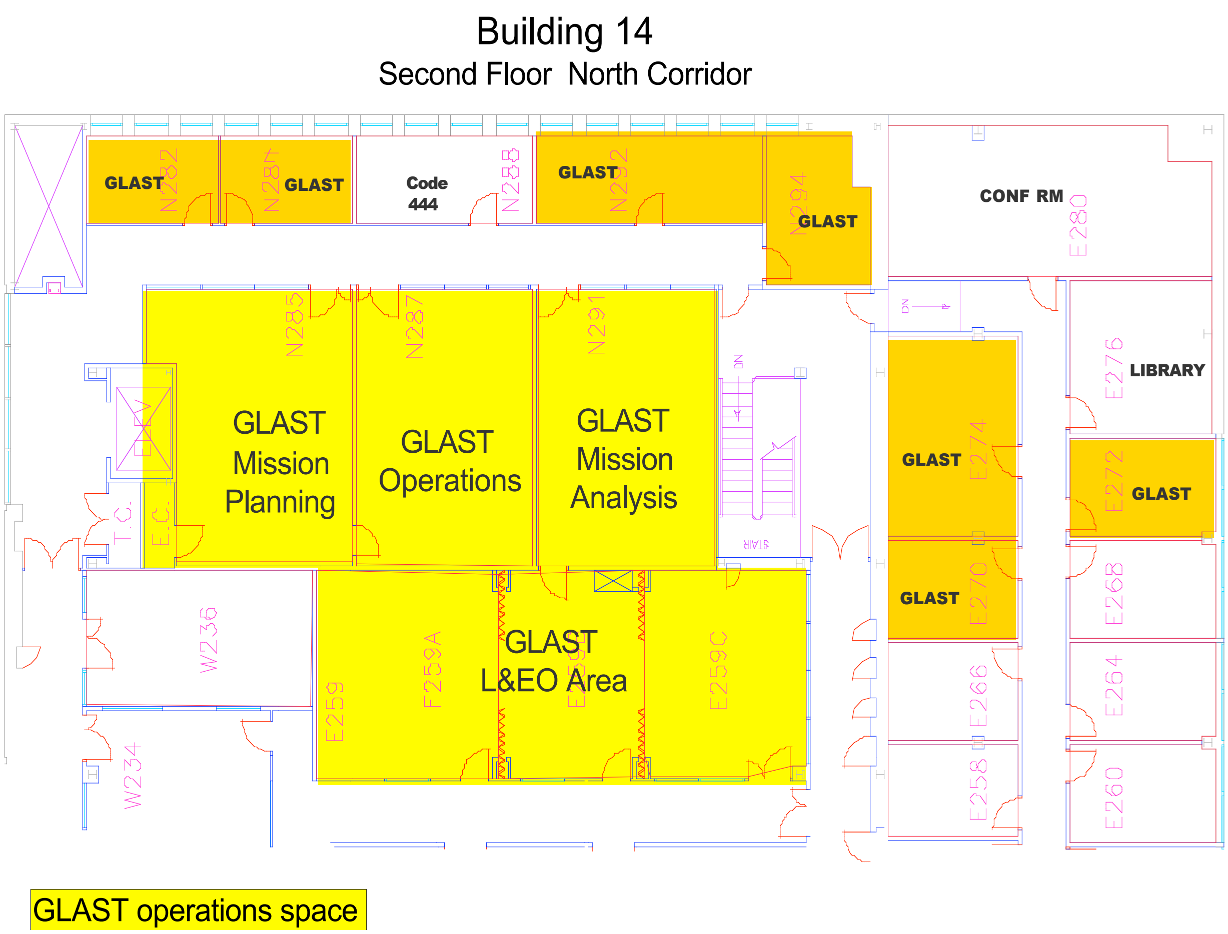


Figure 4.5.1-2. MOC Facility

4.5.2 MOC Operations Control Room Description

The MOC Operations Control Room (OCR) contains the MOC computer systems, voice systems, and furniture necessary for the FOT to perform mission operations. The layout of the OCR is depicted in Figure 4.5.2.

TBD

Figure 4.5.2 - GLAST Operations Room

APPENDIX A: Data Dictionary

Data Entity	Description	Type: Input/ Output/ Internal	Source	Destination	Transfer Method	Storage Location/ Type	Characteristics
As-flown Science Timeline	As-executed science observation timeline annotated w/actual times/events	O	Timeline Monitor	Fileserver, Web	File Sharing	Fileserver, Webserver	1/day to GSSC < 7d; 30 day AFTL LoM archive
ATS and RTS Loads	ATS and RTS command loads for uplink	O	MPS	Fileserver, ITOS	File Sharing, FTP	Fileserver	
USN Schedules	USN ground station pass contact schedules	I	USN	Contact Schedule Muxer	FTP	Local/File	Once per schedule update
USN Status Packets	USN ground station status packets indicating link status and stats for each VC	I	USN	ITOS	TCP/IP socket	Fileserver/File	Received at set interval during real-time USN pass
Configmon Report	Configuration monitor violation report	O	Event Delogger	Fileserver	File Sharing	Fileserver, Webserver	Created from filtering event log for Realtime and Offline
Data Accountability Log	CCSDS Transfer Frame statistics information.	O	Frame Accounting Software, FOT	Fileserver, Web	File Sharing	Fileserver, Webserver	Automated and Manual Updates to log. Frame gap, and frame statistics reports.
E-Documentation	Electronic documentation of ops procedures, S/C, Instrument and Ground System Handbooks	Int	Manual gen from Ops team	Webserver		Webserver	
FDF Orbit Data	Orbit solutions calculated by FDF from TDRSS tracking data	I	FDF	STK	FTP	Fileserver	First week of mission only

Data Entity	Description	Type: Input/ Output/ Internal	Source	Destination	Transfer Method	Storage Location/ Type	Characteristics
Frame and Packet Relays	CCSDS Telemetry Frame and Packet data relays	O	ITOS	ITOS, Open Server, External Systems	TCP/IP Socket	-	Initiated as needed
FSW Loads	S/C loads from Spectrum/AZ; Instrument loads from Inst Teams	I	S/C FSW, LAT FSW, GBM FSW Facilities	ITOS	FTP	Open Server	FTP'd to Open Server, pulled into RT ITOS WS
GN Orbit Products	Orbit data for ground station contact acquisition and scheduling	O	STK	USN, MILA, Wallops	FTP	Fileserver	TLEs supplied to ground stations
GRB Alert from TDRSS	GRB Alert fwd to BAP and GIOC. Included in Level 0 Data	I/O	TDRSS	BAP, GOIC	SFTP	Fileserver, Webserver	Process < .5 secs,
Contact Task File	File containing tasks to be performed on each contact.	Int	Realtime AMAC	-	-	Fileserver	
Improved Inter-Range Vector (IIRV)	Orbital Vector	O	STK	SWSI client, Ground stations	FTP	Fileserver	
Integrated Contact Schedule	Consolidated contact schedule for all supports (USN, MILA, Wallops, DAS, WDISC)	O	Contact Schedule Muxer	MPS, Web, AMAC	File Sharing	Fileserver, Webserver	Updated per schedule updates
Integrated Observatory Timeline	Detailed mission timeline of activities and events for s/c, instruments, and ground.	O	MPS	Fileserver, Web	File Sharing	Fileserver, Webserver	Covers next 4 to 7 days. Updated once per weekday.
ITOS Event Log	ITOS event log	O	Event Delogger	Fileserver, SERS	File Sharing, FTP	Fileserver, Webserver	Created during real-time operations and off-line data processing

Data Entity	Description	Type: Input/ Output/ Internal	Source	Destination	Transfer Method	Storage Location/ Type	Characteristics
ITOS Java Client Communication and Data Transfer	Real-time Telemetry Through ITOS Java Display Client	O	ITOS	Remote Spacecraft, Instrument, and Operations Engineers	TCP/IP	-	
ITOS PDB Command Database	PDB version of T&C DB	O	ITOS	MPS	File Sharing	Fileserver	Special Database format used in SMEX.
ITOS PDB Telemetry Database	PDB version of T&C DB	O	ITOS	DTAS	File Sharing	Fileserver	Special Database format used in SMEX
ITOS Proc	ITOS STOL Procedure (Proc)	Int	Fileserver	ITOS	File Sharing	Fileserver	
Limit Report	Limit violation report	O	Event Delogger	Fileserver	File Sharing	Fileserver, Webserver	Created from filtering event log on real-time and offline systems.
Long Term Archive	Tape based archive of all LoM data	O	Data Archiver	Tape	-	Tape	Monthly and Quarterly archive of LoM data.
Wallops Schedules	Wallops ground station pass contact schedules	I	Wallops	Contact Schedule Muxer	FTP	Local/File	Once per schedule update
Wallops TLM Statistics Report Message	Wallops ground station frame statistics messages for each VC	I	Wallops	ITOS	TCP/IP socket	Fileserver/File	Received at set interval during real-time pass
Missing Frames DB	Database of missing or incomplete CCSDS Transfer Frames	O	Frame Accounting Software	Fileserver, Web, ITOS	File Sharing, FTP	Fileserver, Webserver	
NORAD TLEs	NORAD Two-Line Element sets	I	NASA OIG Web site	STK	FTP	Fileserver, Webserver	Nominal mission ops phase; Update every 1-4 days
ODB	Operational T&C DB converted from T&C DB for runtime	O	ITOS	ITOS		Fileserver	

Data Entity	Description	Type: Input/ Output/ Internal	Source	Destination	Transfer Method	Storage Location/ Type	Characteristics
Offline Telemetry	TLM Frame Files, STDN CCSDS telemetry frame files	I	TDRSS, Ground Stations	ITOS	FTP	Fileserver/File	
On-call DB	Listing of personnel contact info and on-call support schedules	Int	Manual SERS input from Ops teams		-		
Orbital Products	Orbital Products such as SAA, ground track, station view periods, etc.	O	STK	MPS, GSSC, Web	File Sharing	Fileserver, Webserver	"Predicted SAA Passes" being sent to webserver
Page Response	Response from paged on-call personnel	I	On-call user	SERS	RF	-	
Paging	Page to PI for GRB Alerts, Assessment results; to Ops staff for GRB Alerts, S/C or Ground System anomaly	O	SERS	On-call user	RF	-	
Paging Criteria	Set parameters/Configuration for a software program to determine if a paging notification request is necessary	Int	Timeline Monitor, AMAC,	-	-	Fileserver/File	
Paging Notification Request/Paging Request	External request to VMOC/SERS to initiate E-page to on-call personnel	O	AMAC, Timeline Monitor	SERS	Email	SERS DB	S/W process request via email
Playback Telemetry Files	TLM Frame Files, STDN CCSDS telemetry frame files (VC1-4,6)	I	TDRSS, ground stations	ITOS	FTP	Fileserver/File	

Data Entity	Description	Type: Input/ Output/ Internal	Source	Destination	Transfer Method	Storage Location/ Type	Characteristics
Planned Observation Timelines	Time-tagged list of scheduled targets (RA, Dec) and survey modes entry/exit (survey mode types)	O	Science Input Processor	Fileserver, Web, Timeline Monitor, Attitude-dependent TDRSS scheduling	FTP	Fileserver, Webserver	Once per week
Product Notification Msg	Message to GSSC listing all products in delivery set	O	Offline AMAC	GSSC	SFTP	-	Once per pass
Level 0 Data Sets	Level-0 processed data sets. Created from VC0-6	O	ITOS	GSSC, LIOC, GIOC, Fileserver	SFTP	Fileserver, Webserver	File per pass per APID; Generate < 60 mins, 90% of data, transfer to GSSC < 30 mins of gen.
RT CMD Log	Log of all real-time commands sent to s/c	O	Event Delogger	Archive	FTP	Fileserver, Webserver	Created from filtering event log
RT GN TLM	Real-time Housekeeping Telemetry STDN CCSDS Transfer Frames	I	SN, USN, MILA, Wallops	ITOS	TCP/IP socket	Fileserver/File	~5-7 passes/day, 8-12 min duration
RT TDRSS TLM	DAS TDRSS telemetry (VC1) CCSDS Transfer Frames	I	SN	ITOS	TCP/IP socket	Fileserver/File	Burst alerts, Instrument emergency alerts, S/C or Instrument housekeeping
SN Orbit Products	Orbit data for TDRSS contact acquisition and scheduling.	O	STK	SWSI	FTP	SWSI WS	IIRV supplied to SN via SWSI (file)
SSR Read Pointer Status	Sequent print of SSR Read Pointer Counters from VC0 data	Int	ITOS	Frame Accounting Software	File Sharing	Fileserver	Created each pass
Station Status Packets	TLM Statistics Report Message, Station Status Packets	I	Wallops, USN, MILA	ITOS	TCP/IP socket	Fileserver/File	

Data Entity	Description	Type: Input/ Output/ Internal	Source	Destination	Transfer Method	Storage Location/ Type	Characteristics
STOL Directives	STOL directives to control ITOS system	Int	FOT, AMAC	ITOS	GUI input, STOL FIFO	-	
Stored CMD Log	Log of commands in all stored command loads sent to s/c	O	MPS	Archive, GSSC		Fileserver, Webserver	1d To GSSC < 7 days; archive LoM
SWSI Alert Log	SWSI output of alert event window	I	SWSI Client	Fileserver, SERS	FTP	Fileserver	
SWSI TDRSS Schedules	TDRSS DAS Schedules and TDRSS WDISC Schedules	I	SWSI Client	Contact Schedule Muxer	File Sharing	Local/File	
T&C DB	TLM & CMD definitions, limits, calibrations, etc.	I	CM System	ITOS		Fileserver, Webserver	
Table and Memory Dumps	S/C or instrument table and memory dump data extracted from r/t or recorded Tlm	O	ITOS	S/C FSW, BAT FSW, NFI FSW Facilities	FTP	Fileserver	
Target DB	Science target database includes target characteristics, scheduling constraints, observation schedule	Int	TAKO	SOC Content Dispatcher, Timeline Monitor	FTP	Local, Fileserver	Once instance per PPST
Target Observation Requests	Science target observation requests for scheduling	I	SOT	TAKO	File Sharing	Target DB	
TDRSS DAS Schedules	TDRSS DAS handover schedules	I	SWSI Client	Contact Schedule Muxer	File Sharing	Local/File	Once per schedule update
TDRSS Segment Stats	TDRSS telemetry frame statistics between each Scheduled contact	O	TDRSS AMAC	-	File Sharing	Fileserver, Webserver	
TDRSS Status Data	TDRSS status data.	I	SWSI Server	SWSI client	TCP/IP socket	Local/Display	Continuously received once per minute

Data Entity	Description	Type: Input/ Output/ Internal	Source	Destination	Transfer Method	Storage Location/ Type	Characteristics
TDRSS Unscheduled Time (TUT) Report	Report from DSMC showing available TDRSS time	I	DSMC	SWSI	TCP/IP	-	
TDRSS WDISC Schedules	TDRSS MA FWD/RTN support schedules	I	SWSI Client	Contact Schedule Muxer	File Sharing	Local/File	Once per schedule update
Telecommands, Real-time Commanding	Command transmission to GLAST (r/t commands including ATS/RTS/memory loads)	O	ITOS	Wallops, USN, MILA, TDRSS	TCP/IP	-	Via GN or TDRSS
Tlm Archive	Archive of all Tlm received via ground stations	O	ITOS	-	-	-	LoM, 30 day online
TLM Frame Files	STDN CCSDS telemetry frame files (VC0-4,6), TDRSS CCSDS Telemetry frame files (VC5)	I	Wallops, USN, MILA, TDRSS	ITOS	FTP	Fileserver/File	Frame file per VC per pass
Trend Data	Trend data calculated from Tlm data	Int	DTAS	-		DTAS Server	Trend points accumulated for each orbit over LoM
Trending & Analysis Products	Reports, plots, tables of Tlm and trend data	O	DTAS		-	-	
Trending User Requests, Plots, and Data	Requests and Trending input and output from the DTAS client or webserver to and from the DTAS server	I,O	DTAS Client, Web	DTAS Server	TCP/IP		
Trending Sequential Print Files	Trending Sequential Print Files containing mnemonic values versus spacecraft time.	O	ITOS	DTAS, Fileserver	FTP, File Sharing	Fileserver	Created when processing TLM Frame files

Appendix B: Internal Interface Definitions

Internal Interface	Description	Source	Destination	Definition Location
ACS messages	Sequential print of BAT equivalent ACS messages.	ITOS	Timeline Monitor	Defined in this appendix.
As-flown Science Timeline	As-executed science observation timeline annotated w/actual times/events	Timeline Monitor	Fileserver, Web	Defined in *GLAST Operations Data Products ICD*, GLAST-OMI-005
ATS and RTS Loads	ATS and RTS command loads for uplink	MPS	Fileserver, ITOS	Defined in TRACE Telemetry & Command Specification
Configmon Report	Configuration monitor violation report	Event Delogger	Fileserver	Defined in this appendix
Data Accountability Log	CCSDS Transfer Frame statistics information.	Frame Accounting Software, FOT	Fileserver, Web	Defined: TBD
Ephemeris	Used to determine orbital constraints for scheduling observations.	STK	GSSC	Defined in this appendix
Frame and Packet Relays	CCSDS Telemetry Frame and Packet data relays	ITOS	ITOS, Open Server, External Systems	Defined in *GLAST Operations Data Products ICD*, GLAST-OMI-005
Contact Task File	File containing tasks to be performed on each contact.	Ground Station and TDRSS AMAC	-	Defined: TBD
Integrated Contact Schedule	Consolidated contact schedule for all supports (Wallops, USN, MILA, DAS, WDISC)	Contact Schedule Muxer	MPS, Web, AMAC	Defined in this appendix.
Integrated Observatory Timeline	Detailed mission timeline of activities and events for s/c, instruments, and ground.	MPS	Fileserver, Web	Defined in this appendix.
ITOS Event Log	ITOS event log	ITOS	Event Delogger, SERS, AMAC	Defined in this appendix.
ITOS PDB Command Database	PDB version of T&C DB	ITOS	MPS	Defined in TRACE Telemetry & Command Specification
ITOS PDB Telemetry Database	PDB version of T&C DB	ITOS	DTAS	Defined in TRACE Telemetry & Command Specification
Limit Report	Limit violation report	Event Delogger	Fileserver	Defined in this appendix.
Long Term Archive	Tape based archive of all LoM data	Tape Archiver	Tape	Defined: TBD

Internal Interface	**Description**	**Source**	**Destination**	**Definition Location**
Planned Observation Timeline	Time-tagged list of scheduled targets (RA, Dec) and survey modes entry/exit (survey mode types)	Science Input Processor	Timeline Monitor	Defined in *GLAST Operations Data Products ICD*
Science Activity Timeline	Time-tagged list of telecommands for pointing or instrument mode change	GSSC	Science Input Processor, MPS	Defined in *GLAST Operations Data Products ICD*
Missing Frames DB	Database of missing or incomplete CCSDS Transfer Frames	Frame Accounting Software	Fileserver, Web, ITOS	Defined: TDB
Orbital Products	Orbital Products such as SAA, ground track, station view periods, etc.	STK	MPS, Web	Defined in this appendix.
Paging Criteria	Set parameters/Configuration for a software program to determine if a paging notification request is necessary	ToO Handler, Timeline Monitor, AMAC, GRB Handler	-	Defined: TBD
Paging Notification Request/Paging Request	External request to VMOC/SERS to initiate E-page to on-call personnel	AMAC, GRB Handler, ToO Handler, Timeline Monitor	SERS	Defined: TBD
Real-Time Command Log	List of real-time commands in the event log	Event Delogger	Fileserver	Defined in this appendix
SSR Read Pointer Status	Sequent print of SSR Read Pointer Counters from VC0 data	ITOS	Frame Accounting Software	Defined: TBD
STOL Commands	STOL directives to control ITOS system	FOT, AMAC	ITOS	Defined using DBX format which is in the ITOS documentation.
T&C DB	TLM & CMD definitions, limits, calibrations, etc.	CM System	ITOS	Defined in TRACE Telemetry & Command Specification
TDRSS Segment Stats	TDRSS telemetry frame statistics between each Scheduled contact	TDRSS AMAC	-	Defined: TBD
Tlm Archive	Archive of all Tlm received via ground stations	ITOS	-	Defined: TBD
Trend Data	Trend data calculated from Tlm data	DTAS	-	Defined: TBD
Trending & Analysis Products	Reports, plots, tables of Tlm and trend data	DTAS		Defined: TBD
Trending User Requests, Plots, and Data	Requests and Trending input and output from the DTAS client or webserver to and from the DTAS server	DTAS Client, Web	DTAS Server	Defined: TBD
Trending Sequential Print Files	Trending Sequential Print Files containing mnemonic values versus spacecraft time.	ITOS	DTAS, Fileserver	Defined: TBD

Configmon Report

This section describes the format of the Configmon Report file output from the Event Delogger. The Configmon Report is a subset of an ITOS Event Log file. This report only contains lines that contain configuration type events that are specified by the search criteria, which can be modified in the Event Delogger configuration file.

Filename Convention

The filename is the ITOS event log filename (see ITOS Event Log Definition) prefixed by the letters 'CFG'. For example:

```
CFGGLASTEVT_02-263-2106.TMP
```

File Format

The file begins with header lines that describe how the file was created. They include: report title, full pathname of report, time created, ITOS event file name, type (TDRSS or offline) and the search criteria used to select the events. Following the header is either the ITOS events extracted (for format see the ITOS Event Log Definition) or a text message indicating that no events matched the search criteria.

Sample Configmon Report File

```
Configuration Monitor Violations
File: /home/csmith/moc/my_filesvr/ITOS/output/reports/CfgArchive/offline/CFGGLASTEVT_02-263-2106.TMP
Created: 2003-009-14:38:41
Event file: /home/csmith/test/ed/GLASTEVT_02-263-2106.TMP  Type: offline
Search criteria: dsp_evtlog:|CFG_ALERT
________________________________________________________________________________

00 02-263-21:06:59 NULL_EVENT: dsp_evtlog: opened (append) log file /usr/tcw.GLAST/logs/GLASTEVT_02-263-2106.TMP
29 02-263-21:09:05  CFG_ALERT: configuration tgnormal: "OMTESTU8 isn't ZERO" at 34-306-11:17:00.01500
29 02-263-21:09:10  CFG_ALERT: configuration tgnormal: "OMTESTI32 is less than ZERO" at 34-306-11:17:00.01500
29 02-263-21:09:10  CFG_ALERT: configuration tgnormal: "OMTESTI16 is less than ZERO" at 34-306-11:17:00.01500
29 02-263-21:09:10  CFG_ALERT: configuration tgnormal: "OMTESTI8 is less than ZERO" at 34-306-11:17:00.01500
29 02-263-21:09:10  CFG_ALERT: configuration tgnormal: "OMTESTI32 isn't ZERO" at 34-306-11:17:00.01500
29 02-263-21:09:10  CFG_ALERT: configuration tgnormal: "OMTESTI16 isn't ZERO" at 34-306-11:17:00.01500
00 02-263-21:12:24 NULL_EVENT: dsp_evtlog: closing log file /usr/tcw.GLAST/logs/GLASTEVT_02-263-2106.TMP
Event Delog complete.
```

Integrated Contact Schedule

This section describes the format of the Integrated Contact Schedule. This file is the output of the Contact Schedule Muxer. It is input to the Mission Planning System (MPS) software and the AMAC software. It is also stored on the MOC file server.

The file may contain several days' worth of scheduled contacts. .

File naming convention

`ICS_yyyy_ddd_hhmmss_vv.txt`

where `yyyy` is the 4-digit year
`ddd` is the 3-digit day of year (001-366)
`hhmmss` is the hours, minutes and seconds
`vv` is a version number. Initially, 00 and incremented if the contact schedule muxer makes additional runs for the same start time.

The date indicates the start time of the schedule (not necessarily the first record in the file). The file will contain all known ground station and TDRSS scheduled contacts and DAS handovers after the start time.

Filename example: `ICS_2001_355_123510_00.txt`

File layout

Fields are pipe delimited. Comment lines start with #. Blank lines are permitted.

Column Heading	Description
Begin	Time of the start of the pass (AOS) Format: yyyy/ddd/hh:mm:ss
End	Time of the end of the pass (LOS) Format: yyyy/ddd/hh:mm:ss
Duration	Duration of the contact (HH:MM:SS)
Station	Station id
Rate	Down-link data rate, Kb/sec
Pass	Ground station pass number For TDRSS, use "na"
Service	Service type Valid values for TDRSS: MAF or MAR

Column Heading	Description
	Valid values for USN: na
Antenna	For ground stations, this is the antenna id number For TDRSS, this indicates the schedule type: Use DAS for a handover record under Demand Access System Use WDISC for a scheduled contact
Config	Configuration notes (optional)

Integrated Contact Schedule Sample

```
#             Integrated Contact Schedule: ICS_2002_219_000000_12.txt
#             Start time:   2002/219/00:00:00
#             Stop time:    2002/220/06:40:00
#             Files Merged: /filesvr/dev_mocfilesvr/planning/schedules/current/MCS_2002_219_12.txt
#                           /filesvr/dev_mocfilesvr/planning/schedules/current/NORM_NCC_3782_asf_2002219015940.txt
#
#                                                           Rate
#   Begin           |     End           | Duration | Stn |  (kbps)  | Pass   | Service | Ant    | Config
2002/219/00:46:00 | 2002/219/03:32:00 | 02:46:00 | TDW |       2 |     na | MAF      | WDISC | EventID=0011870 SUPIDEN=A3782MS
Service=1
2002/219/00:46:00 | 2002/219/03:32:00 | 02:46:00 | TDW |       4 |     na | MAR      | WDISC | EventID=0011870 SUPIDEN=A3782MS
Service=2
2002/219/02:00:00 | 2002/219/02:08:00 | 00:08:00 | MAL |    2254 |    124 | na       | 02     |
2002/219/02:00:00 | 2002/219/02:30:00 | 00:30:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0011830 SUPIDEN=A3782MS
Service=1
2002/219/02:00:30 | 2002/219/02:30:00 | 00:29:30 | TDW |       8 |     na | MAR      | WDISC | EventID=0011830 SUPIDEN=A3782MS
Service=2
2002/219/03:30:00 | 2002/219/03:40:00 | 00:10:00 | MAL |    2254 |    125 | na       | 01     |
2002/219/05:00:00 | 2002/219/05:07:00 | 00:07:00 | MAL |    2254 |    126 | na       | 01     |
2002/219/06:19:00 | 2002/219/06:25:00 | 00:06:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012300 SUPIDEN=A3782MS
Service=1
2002/219/06:26:00 | 2002/219/06:32:00 | 00:06:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012310 SUPIDEN=A3782MS
Service=1
2002/219/06:30:00 | 2002/219/06:38:00 | 00:08:00 | MAL |    2254 |    127 | na       | 02     | notes
2002/219/08:00:00 | 2002/219/08:09:00 | 00:09:00 | MAL |    2254 |    128 | na       | 01     |
2002/219/09:23:00 | 2002/219/09:33:00 | 00:10:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012010 SUPIDEN=A3782MS
Service=1
2002/219/09:28:00 | 2002/219/09:46:00 | 00:18:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012040 SUPIDEN=A3782MS
Service=1
2002/219/09:30:00 | 2002/219/09:38:00 | 00:08:00 | MAL |    2254 |    129 | na       | 01     |
2002/219/09:38:00 | 2002/219/09:48:00 | 00:10:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012020 SUPIDEN=A3782MS
Service=1
2002/219/12:00:00 | 2002/219/12:30:00 | 00:30:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012380 SUPIDEN=A3782MS
Service=1
2002/219/12:00:30 | 2002/219/12:30:00 | 00:29:30 | TDW |       8 |     na | MAR      | WDISC | EventID=0012380 SUPIDEN=A3782MS
Service=2
2002/219/12:10:00 | 2002/219/12:40:00 | 00:30:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012430 SUPIDEN=A3782MS
Service=1
2002/219/12:10:30 | 2002/219/12:40:00 | 00:29:30 | TDW |       8 |     na | MAR      | WDISC | EventID=0012430 SUPIDEN=A3782MS
Service=2
2002/219/12:30:00 | 2002/219/12:39:00 | 00:09:00 | MAL |    2254 |    131 | na       | 01     |
2002/219/13:00:00 | 2002/219/13:30:00 | 00:30:00 | TDW |       1 |     na | MAF      | WDISC | EventID=0012390 SUPIDEN=A3782MS
Service=1
```

```
2002/219/13:00:30 | 2002/219/13:30:00 | 00:29:30 | TDW |      8 |    na | MAR      | WDISC | EventID=0012390 SUPIDEN=A3782MS
Service=2
2002/219/14:00:00 | 2002/219/14:12:00 | 00:12:00 | MAL |   2254 |   132 | na       | 01    |
2002/219/15:30:00 | 2002/219/15:40:00 | 00:10:00 | MAL |   2254 |   133 | na       | 02    |
2002/219/17:00:00 | 2002/219/17:08:00 | 00:08:00 | MAL |   2254 |   134 | na       | 01    |
2002/219/18:30:00 | 2002/219/18:37:00 | 00:07:00 | MAL |   2254 |   135 | na       | 01    |
2002/219/23:00:00 | 2002/219/23:07:00 | 00:07:00 | MAL |   2254 |   138 | na       | 01    |
2002/220/00:30:00 | 2002/220/00:38:00 | 00:08:00 | MAL |   2254 |   139 | na       | 02    |
2002/220/02:00:00 | 2002/220/02:09:00 | 00:09:00 | MAL |   2254 |   140 | na       | 01    |
2002/220/03:30:00 | 2002/220/03:40:00 | 00:10:00 | MAL |   2254 |   141 | na       | 01    |
2002/220/06:30:00 | 2002/220/06:40:00 | 00:10:00 | MAL |   2254 |   143 | na       | 01    |
```

Integrated Observatory Timeline

This section describes the format of the Integrated Observatory Timeline file output from the Mission Planning System (MPS). This is the same file as Integrated Print file in the legacy MPS software. The Integrated Observatory Timeline file is a text file that contains a tabular listing of the events and stored commands that are scheduled to occur during the specified planning period.

Filename Convention

`IOTL_yyyyddd_hhmmss_loadname.txt`

Where `yyyyddd_hhmmss` is the time of the first command in the load

`loadname` is the name used in MPS to identify the load. The MPS user may enter any useful name. MPS defaults the name to: `SCSdddbv`

Where SCS is the onboard processor name

- ddd the day of year
- b the ATS buffer for the load.
- v sequence number (initially 1) and is incremented, if several loads are defined for the day.

Note that the load name is not the same as the ATS load file name.

Examples:

```
IOTL_2002171_164325_SCS171B1.txt
IOTL_2002172_000000_SCS172A2.txt
```

File Format

The Integrated Observatory Timeline file is composed of three sections:
1. Header
2. Detail List
3. Load continuity

1. Header Section
The header section contains summary information about a stored command load in the selected planning period when the report was created. The information is fairly self explanatory. See the sample file.

2. Detail List Section
The detail list section contains time-ordered list of the stored commands and events that are scheduled to occur during the planning period. For continuity, the list begins with the

commands from the previous prime load, and ends with commands from the following prime load. The table columns are:
EVENT: For events, the name of a predicted event such as SAA Entry or Ascending Node. Blank for commands
TIME: The date and time stamp of the command or event. Format is YY DDD/HH:MM:SS
CMD #: For commands, this column contains the sequence number of the command within the load. Blank for events.
CRIT: For commands, Y if the command is a critical command as indicated in the command database, N otherwise. Blank for events.
COMMAND: For commands, the command mnemonic. Blank for events.
SUBMNEMONICS: For commands, the command submnemonics in a comma separated list. The submnemonic list will wrap to the next line if necessary. Blank for events.
SEQUENCE DURATION: The duration of an activity.
ATS BUFF: For commands, the ATS buffer the command is stored in (A or B). Blank for events.
TRIG/ACT/RTS NAME: This column can contain one of the following:

The event trigger that caused the command to be scheduled.
The name of the ATS activity definition containing the command
The name of the RTS activity definition containing the command

TRIGGER/ACTIVITY/EVENT/RTS DESCRIPTION: The comment field of the trigger, activity, event, or RTS involved with the command or event.

3. Load continuity
This section contains information about the preceding and succeeding loads.

Sample File

```
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                                                      GLAST Release 1 INTEGRATED PRINT
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
LOAD NAME: SCS330B2                                      LOAD PERIOD START: 2001 330/00:00:00                    TIME FIRST ATS COMMAND: 2001 330/00:00:04
LOAD GENERATION ERRORS: 0                                LOAD PERIOD STOP:  2001 330/23:59:59                    TIME LAST  ATS COMMAND: 2001 330/23:59:59
CHECKSUM: 00000fd0                                       TIME LOAD   GENERATED: 2001 319/21:52:48                BUFFER AUTOSWITCH: Yes
LOAD SIZE: 114                                            TIME REPORT GENERATED: 2001 319/21:55:05               TIME BUFFER SWITCH COMMAND: 2001 330/23:59:59
BYTES FREE IN BUFFER: 34886                              LOAD STATUS: Alternate TYPE: Command                    TIME FIRST RTS COMMAND: No RTS Commands
NUMBER OF LOAD COMMANDS: 1                               PREVIOUS LOAD: SCS329A1                                 TIME LAST  RTS COMMAND: No RTS Commands
NUMBER OF CRITICAL COMMANDS: 0                          NEXT LOAD:      No Next Prime Load                       FIRST COMMAND NUMBER: 1
ATS LOAD UPLINK WINDOW START: 2001 329/00:00:00        PREV LOAD BUFFER: A                                       LAST  COMMAND NUMBER: 5
ATS LOAD UPLINK WINDOW STOP:  2001 330/00:00:04        THIS LOAD BUFFER: B                                       NUMBER OF RTS TABLES USED: 0
TOTAL COMMANDS: 5                                       NEXT LOAD BUFFER:                                        ORBIT VECTORS: None

SELECT COMMAND: SMTBLSELECT TABLEID=65,SOURCE_TABLE=SRCZERO,DEST_TABLE=DESTSRAM                                     FOT: : :  [01-319-16:52:48]
COMMIT COMMAND: SMTBLCOMMIT SELECT=CKENABLE,CHECKSUM=4048                                                           FOT: : :  [01-319-16:52:48]
OUTPUT FILES:                                            INPUT FILES:
T01330SLTN65.00 2001 319/21:52:48                         prime.con 2001 318/08:54:21
T01330SCTN65.00 2001 319/21:52:48                         prime.evt 2001 318/15:44:05
                                                          triggers 2001 318/17:24:29
                                                          dictionary 2001 319/10:19:55
                                                          cdb.current 2001 318/09:05:26
                                                           PPST_20013291334_20013301200_01.txt
RTS LOADS REQUIRED FOR UPLINK
LOAD NAME                     UPLINK WINDOW
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Error Messages:
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Informational/Warning Messages:
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                                                        GLAST Release 1 INTEGRATED PRINT
                            CMD                                                                                     SEQUENCE  ATS  TRIG/ACT/RTS  TRIGGER/ACTIVITY/EVENT/RTS
EVENT        TIME            #   CRIT COMMAND      SUBMNEMONICS                                                   DURATION  BUFF NAME           DESCRIPTION
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
              01 329/16:34:15  4      N  FOPPTREQUEST OBSERVATION_ID=0x02000135,MERIT=0.5,RA=123.45,                     A    TargetTrigger
                                                        DEC=15.23,ROLL=105.1,BAT_MODE=0x0000,XRT_MODE=1,
                                                     UVOT_MODE=2
              01 329/17:34:15  5      N  FOPPTREQUEST OBSERVATION_ID=0x03000134,MERIT=0.99,RA=23.45,                     A    TargetTrigger
                                                        DEC=-5.23,ROLL=5.1,BAT_MODE=0x0000,XRT_MODE=1,
                                                     UVOT_MODE=2
              01 329/18:34:15  6      N  FOPPTREQUEST OBSERVATION_ID=0x01000130,MERIT=0,RA=0,DEC=0,                      A    TargetTrigger
                                                        ROLL=0,BAT_MODE=0x0000,XRT_MODE=0,UVOT_MODE=0
              01 329/19:00:00  7      N  FOPPTREQUEST OBSERVATION_ID=0x02000130,MERIT=0,RA=0,DEC=0,                      A    TargetTrigger
                                                        ROLL=0,BAT_MODE=0x0000,XRT_MODE=0,UVOT_MODE=0
            01 329/23:59:59  8     N  SCATSSWITCH                                                                        A
            01 330/00:00:04  1     N  BATNOOP_HI                                                                         B
            01 330/00:00:06  2     N  BATNOOP_LO                                                                         B
```

```
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                                                      GLAST Release 1 INTEGRATED PRINT
                                           CMD                                                                                              SEQUENCE   ATS   TRIG/ACT/RTS   TRIGGER/ACTIVITY/EVENT/RTS
EVENT          TIME                        #    CRIT COMMAND       SUBMNEMONICS                                                            DURATION  BUFF NAME            DESCRIPTION
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
               01 330/00:00:08  3      N  BATBCNOOP                                                                                  B
AscendingNod  01  330/01:32:44
SAA ENT        01  330/03:01:14
SAA EXIT       01  330/03:06:19
AscendingNod  01  330/03:09:34
SAA ENT        01  330/04:36:03
SAA EXIT       01  330/04:46:15
AscendingNod  01  330/04:46:24
SAA ENT        01  330/06:10:52
AscendingNod  01  330/06:23:14
SAA EXIT       01  330/06:26:02
SAA ENT        01  330/07:45:44
AscendingNod  01  330/08:00:04
SAA EXIT       01  330/08:03:02
SAA ENT        01  330/09:22:48
AscendingNod  01  330/09:36:54
SAA EXIT       01  330/09:38:01
SAA ENT        01  330/11:01:05
SAA EXIT       01  330/11:11:35
AscendingNod  01  330/11:13:45
                 01 330/12:00:00   4       N   FOPPTREQUEST  OBSERVATION_ID=0x02000130,MERIT=0,RA=0,DEC=0,                  B    TargetTrigger
                                                              ROLL=0,BAT_MODE=0x0000,XRT_MODE=0,UVOT_MODE=0
AscendingNod  01  330/12:50:35
AscendingNod  01  330/14:27:25
SAA ENT        01  330/15:18:12
SAA EXIT       01  330/15:25:25
AscendingNod  01  330/16:04:15
SAA ENT        01  330/16:52:28
SAA EXIT       01  330/17:04:42
AscendingNod  01  330/17:41:05
SAA ENT        01  330/18:26:51
SAA EXIT       01  330/18:44:07
AscendingNod  01  330/19:17:56
SAA ENT        01  330/20:02:59
SAA EXIT       01  330/20:19:57
AscendingNod  01  330/20:54:46
SAA ENT        01  330/21:41:53
SAA EXIT       01  330/21:55:33
AscendingNod  01  330/22:31:36
SAA ENT        01  330/23:23:38
SAA EXIT       01  330/23:28:26
               01 330/23:59:59  5      N  SCATSSWITCH                                                                                B

--------------------------------------------------------------------------------
                        GLAST Release 1 LOAD CONTINUITY REPORT
  Between   Load:   SCS329A1   Buffer: A
       and  Load:   SCS330B2   Buffer: B
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
                        GLAST Release 1 LOAD CONTINUITY REPORT
  Between   Load:   SCS330B2   Buffer: B
       and  Load:              Buffer:
```

ITOS Event Log

This describes the format of the Event Log file output from the Integrated Test and Operations System (ITOS). The Event Log file is a capture of events that the ITOS has processed. The Event Log file is a text file that contains a space delimited listing of event data in a time ordered sequence.

Filename Convention

GLASTEVT_YY-DOY-HHMM.TMP
where

YY-DOY-HHMM : is the date and time when the frame file was created.
YY is the 2-digit year
DOY is the day of year
HHMM is the time in hours and minutes

Example
GLASTEVT_02-064-1331.TMP

File Format
The ITOS Event Log file is composed of five columns:

1. Escape sequence: To define event color
   See: http://itos.gsfc.nasa.gov/ITOS/event/Event_messages.html

2. Event Code : A two digit code used to numerically define the event.
   See: http://itos.gsfc.nasa.gov/ITOS/EvtMsg/Event_Types.html

3. Time : Time of event YY-DOY-HH:MM:SS

4. Enumerated Event Type : An event mnemonic.
   See: http://itos.gsfc.nasa.gov/ITOS/EvtMsg/Event_Types.html

5. Text : A text message describing the action of the event.

Sample File

00 02-031-20:44:29 NULL_EVENT: dsp_evtlog: opened log file /usr/tcw.GLAST/logs/startup_log during startup
05 02-031-20:44:31 TM_MSG: invoked listening for connection on port 33000
17 02-031-20:44:33 STOL_MSG: Initializing command subsystem………..

Limit Report

This section describes the format of the Limit Report file output from the Event Delogger. The Limit Report is a subset of an ITOS Event Log file. This report only contains lines that with limit type events that are specified by the search criteria. The search criteria can be modified in the Event Delogger configuration file.

Filename Convention

The filename is the ITOS event log filename (see ITOS Event Log Definition) prefixed by the letters 'LIM'. For example:

LIMGLASTEVT_02-263-2056.TMP

File Format

The file begins with header lines that describe how the file was created. They include: report title, full pathname of report, time created, ITOS event file name, type (DAS, RT, or offline) and the search criteria used to select the events. Following the header is either the ITOS events extracted (for format see the ITOS Event Log Definition) or a text message indicating that no events matched the search criteria.

Sample Limit Report File

```
Limit Violations
File: /home/csmith/moc/my_filesvr/ITOS/output/reports/LimArchive/tdrss/LIMGLASTEVT_02-263-2056.TMP
Created: 2003-009-14:37:48
Event file: /home/csmith/test/ed/GLASTEVT_02-263-2056.TMP  Type: tdrss
Search criteria: dsp_evtlog:|RED_VIOL|YEL_VIOL|IN_LIMITS
________________________________________________________________________________________

00 02-263-20:56:15 NULL_EVENT: dsp_evtlog: opened (append) log file /usr/tcw.GLAST/logs/GLASTEVT_02-263-2056.TMP
01 02-263-20:58:11   RED_VIOL: Red    high violation OMTESTU32EU cnv = 405841 at 34-306-11:13:39.01500
02 02-263-20:58:11   YEL_VIOL: Yellow low  violation OMTESTI8EU cnv = -48766.5 at 34-306-11:13:39.01500
01 02-263-20:58:11   RED_VIOL: Red    low  violation OMTESTI16EU cnv = -1.332157113e+12 at 34-306-11:13:39.01500
01 02-263-20:58:11   RED_VIOL: Red    low  violation OMTESTI32EU cnv = -3.057152869e+26 at 34-306-11:13:39.01500
01 02-263-20:58:11   RED_VIOL: Red    low  violation OMTESTFLOAT32EU cnv = -3.271828684e+29 at 34-306-11:13:39.01500
04 02-263-20:59:13  IN_LIMITS: Parameter  in limits OMTESTI32 raw = -18999 at 34-306-11:14:41.01500
04 02-263-20:59:13  IN_LIMITS: Parameter  in limits OMTESTFLOAT32 raw = -12342 at 34-306-11:14:41.01500
04 02-263-20:59:13  IN_LIMITS: Parameter  in limits OMTESTFLOAT64 raw = -2343.234255 at 34-306-11:14:41.01500
04 02-263-20:59:13  IN_LIMITS: Parameter  in limits OMTESTU16EU cnv = -5930.5 at 34-306-11:14:41.01500
00 02-263-21:02:38 NULL_EVENT: dsp_evtlog: closing log file /usr/tcw.GLAST/logs/GLASTDEV_02-263-2056.TMP
Event Delog complete.
```

Orbital Products

This section describes the format of the STK flight dynamics products input to the MPS. The products are classified as Duration Events, Orbital Events, View Periods, and spacecraft ephemerides. The file format that MPS currently expects is a slightly different format from the products that STK can produce so the STK Automation software performs the conversion.

The following sections describe the files in each classification. First the STK format is given, then the MPS format is given along with a description of the processing needed to convert from the STK format into the MPS format.

In all file naming conventions:

yyyy is the 4-digit year of the date that the file was created
ddd is the 3-digit day of the year
vv is a version number (initial version is 00) it is incremented for subsequent version of the file generated for the same date

Duration Events

Duration events are events that have a start and stop time.

STK product description

STK produces a file containing the information for each of these events.
1. Eclipse
2. South Atlantic Anomaly
3. other events are important to GLAST.

MPS does not require any particular events to be produced.

File naming convention:
1. Eclipse file: GyyyydddSUN.vv
2. SAA file: GyyyydddSAA.vv

File format:

Line 1: Field Heading - see sample
Line 2 and so forth – the event record. Comma-delimited fields.

Field 1: Start time of the event.
Field 2: The orbit number in which the event starts (quotes are optional)
Field 3: Stop time of the event
Field 4: The orbit number in which the event stops
Field 5: Duration in minutes of the event (decimal)

All time formats are DOY/ YYYY HH:MM:SS.ss

Sample output of STK:

```
"Start Time (UTCJ)","Orbit Start","Stop Time (UTCJ)","Orbit Stop","Duration (min)"
336/2001 00:00:00.00,"14599",336/2001 00:12:14.72,"14600",12.245
336/2001 00:32:48.76,"14600",336/2001 01:49:04.94,"14601",76.270
336/2001 02:09:39.65,"14601",336/2001 03:25:55.10,"14602",76.257
336/2001 03:46:30.55,"14602",336/2001 05:02:45.25,"14603",76.245
336/2001 05:23:21.40,"14603",336/2001 06:39:35.44,"14604",76.234
```

MPS input description

The duration events input file contains the data for all of the duration events for the planning period. The STK Automation script concatenates the STK event files. It adds a line that identifies the event type and copies the corresponding STK product.

File naming convention:
GyyyydddDUREVT.vv

File format:

For each event:
Line 1: Event name
Line 2: Heading (copied from the STK file) - see sample
Line 3 and so forth (copied from the STK file) – the event record. Comma-delimited fields.
Field 1: Start time of the event.
Field 2: The orbit number in which the event starts (quotes are optional)
Field 3: Stop time of the event
Field 4: The orbit number in which the event stops
Field 5: Duration in minutes of the event (decimal)
Line N: blank line before the next event section

Sample input to MPS:

File name: S2001366DUREVT.00

```
ECLIPSE
"Start Time (UTCJ)","Orbit Start","Stop Time (UTCJ)","Orbit Stop","Duration (min)"
336/2001 00:00:00.00,"14599",336/2001 00:12:14.72,"14600",12.245
336/2001 00:32:48.76,"14600",336/2001 01:49:04.94,"14601",76.270
336/2001 02:09:39.65,"14601",336/2001 03:25:55.10,"14602",76.257
336/2001 03:46:30.55,"14602",336/2001 05:02:45.25,"14603",76.245
336/2001 05:23:21.40,"14603",336/2001 06:39:35.44,"14604",76.234

South-Atlantic-Anomaly
"Start Time (UTCJ)","Orbit Start","Stop Time (UTCJ)","Orbit Stop","Duration (min)"
336/2001 03:05:33.90,"14601",336/2001 03:10:52.81,"14601",5.315
336/2001 04:40:23.02,"14602",336/2001 04:50:48.75,"14602",10.429
336/2001 06:15:13.46,"14603",336/2001 06:30:36.98,"14604",15.392
336/2001 07:50:05.94,"14604",336/2001 08:07:24.13,"14605",17.303
336/2001 09:27:20.04,"14605",336/2001 09:42:23.37,"14606",15.055
336/2001 11:05:37.95,"14606",336/2001 11:15:47.03,"14606",10.151
```

Orbital Events

The orbital events are instantaneous events that pertain to an orbit. These events have just a start time as opposed to the duration events which also have an end time. The orbital events are:
1. Ascending Node
2. Descending Node
3. Apogee
4. Perigee

No reformatting of the orbital events file is required. MPS uses the same format that is output by STK.

File naming convention:
GyyyydddORBEVT.vv

File format

Line 1: Heading - see sample (the heading is all on one line)
Line 2 and so forth – the event record. Comma-delimited fields.
- Field 1: The orbit number
- Field 2: Orbit start time
- Field 3: Ascending node time (typically the same as the orbit start time)
- Field 4: Longitude (degrees) of the ascending node
- Field 5: Apogee time
- Field 6: Altitude (kilometers) of the apogee
- Field 7: Orbit end time
- Field 8: Descending node time (typically the same as the orbit start time)
- Field 9: Longitude (degrees) of the Descending node
- Field 10: Perigee time
- Field 11: Altitude (kilometers) of the perigee

All time formats are DOY/ YYYY HH:MM:SS.ss

STK apparently produced the "Not in Pass" entries. It happens on the first and last event records. These are not required, but MPS can handle them if they occur in a file.

Example file:

File name: S2001366ORBEVT.00

```
"Orbit Number","Start Time (UTCJ)","Time of Ascen Node (UTCJ)","Lon Ascen Node
(deg)","Time of Apogee (UTCJ)","Apogee (km)","End Time (UTCJ)","Time of Descen Node
(UTCJ)","Lon Descen Node (deg)","Time of Perigee (UTCJ)","Perigee (km)"
14599,336/2001 00:00:00.00,Not in Pass,Not in Pass,Not in Pass,Not in Pass,336/2001
00:00:17.37,Not in Pass,Not in Pass,Not in Pass,Not in Pass
14600,336/2001 00:00:17.37,336/2001 00:00:17.37,83.729,336/2001
01:35:36.01,626.746194,336/2001 01:37:08.15,336/2001 00:48:41.50,-108.371,336/2001
00:45:48.94,589.655902
14601,336/2001 01:37:08.15,336/2001 01:37:08.15,59.517,336/2001
03:12:21.63,626.758477,336/2001 03:13:58.89,336/2001 02:25:32.19,-132.583,336/2001
02:22:30.42,589.675043
14602,336/2001 03:13:58.89,336/2001 03:13:58.89,35.306,336/2001
04:49:07.49,626.837215,336/2001 04:50:49.56,336/2001 04:02:22.81,-156.794,336/2001
03:59:12.14,589.639842
```

View Periods

The View Periods event file contains the predicted periods that the spacecraft is in view of the ground station. It contains the view period information for all the applicable stations in one file. The file contains several days of predicted view periods. The MPS uses the view period information along with the schedule contact data to create Pass events. The MPS operator may use the Pass event to schedule ATS commands.

STK product description

STK produces a view period file for each station and antenna.

File naming convention

GyyyydddsssssVPD.vv
where sssss is a 5-character id for the station and antenna.

File format

Line 1: Heading - see sample
Line 2 and so forth – the event record. Comma-delimited fields.
- Field 1: Start time of the event.
- Field 2: The orbit number in which the view period starts (quotes are optional)
- Field 3: Stop time of the view period
- Field 4: The orbit number in which the view period stops
- Field 5: Duration in minutes of the view period (decimal)
- Field 6: Time of the maximum elevation
- Field 7: Maximum elevation in kilometers

All time formats are DOY/ YYYY HH:MM:SS.ss

Example output from STK:

File name: S2001322AG129VPD.00

```
"Start Time (UTCJ)","Orbit Start","Stop Time (UTCJ)","Orbit Stop","Duration
(min)","Max Elev Time (UTCJ)","Max Elevation (deg)"
336/2001 04:45:32.70,"14602",336/2001 04:57:16.89,"14603",11.736,336/2001
04:51:25.71,20.771
336/2001 06:21:22.27,"14603",336/2001 06:32:32.77,"14604",11.175,336/2001
06:26:57.09,16.168
336/2001 16:51:50.36,"14610",336/2001 17:03:26.82,"14610",11.608,336/2001 16:57:38.67,23.771
```

MPS input description

To product the view period file for MPS, the STK Automation script adds the station and antenna id (Line 1 below) and concatenates the individual station view period files into one file.

File naming convention

GyyyydddVIEWPD.vv

File format

For each station:
Line 1: STN/aa where STN is 3-character station id and aa is a two-digit antenna number.
Line 2: Heading (copied from the STK file) - see sample
Line 3 and so forth (copies event records from the STK file Comma-delimited fields.

Field 1: Start time of the event.
Field 2: The orbit number in which the view period starts (quotes are optional)
Field 3: Stop time of the view period
Field 4: The orbit number in which the view period stops
Field 5: Duration in minutes of the view period (decimal)
Field 6: Time of the maximum elevation
Field 7: Maximum elevation in kilometers

Line N: blank line before the next event section

All time formats are DDD/ YYYY HH:MM:SS.ss

Example file for MPS:

The sample shows a case where two of STK's view period files, S2001366AG129VPD.00 and S2001366MCM99VPD.00, were combined to create the MPS view period file.

File name: S2001366VIEWPD.00

```
AG1/29
"Start Time (UTCJ)","Orbit Start","Stop Time (UTCJ)","Orbit Stop","Duration
(min)","Max Elev Time (UTCJ)","Max Elevation (deg)"
336/2001 04:45:32.70,"14602",336/2001 04:57:16.89,"14603",11.736,336/2001
04:51:25.71,20.771
336/2001 06:21:22.27,"14603",336/2001 06:32:32.77,"14604",11.175,336/2001
06:26:57.09,16.168
336/2001 16:51:50.36,"14610",336/2001 17:03:26.82,"14610",11.608,336/2001
16:57:38.67,23.771

MCM/99
"Start Time (UTCJ)","Orbit Start","Stop Time (UTCJ)","Orbit Stop","Duration
(min)","Max Elev Time (UTCJ)","Max Elevation (deg)"
336/2001 01:07:21.05,"14600",336/2001 01:18:11.39,"14600",10.839,336/2001
01:12:13.77,49.611
336/2001 02:43:11.74,"14601",336/2001 02:54:10.07,"14601",10.972,336/2001
02:47:49.08,65.461
336/2001 04:18:51.44,"14602",336/2001 04:29:58.03,"14602",11.110,336/2001
04:23:42.34,75.538
336/2001 05:55:12.23,"14603",336/2001 06:06:23.89,"14603",11.194,336/2001
06:00:03.22,38.334
```

Spacecraft Ephemeris

Input to the Load EPV Vector capability. STK Automation filters the SCACS_EPH* file to produce 20 or so vectors per day. EPV stands for Extended Precision Vector. Each record contains the predicted position and velocity.

Ephemeris file format

Line 1: skip (field titles)
Line 2-n: Vector records (comma delimited)

Field	Description	Units
1	UTC in STK format	ddd/yyyy HH:MM:SS.ss
2	X position (J2000 ECI)	meters
3	Y position (J2000 ECI)	meters
4	Z position (J2000 ECI)	meters
5	X velocity	meters/second
6	Y velocity	meters/second
7	Z velocity	meters/second

Sample EPV file:

```
"Time (UTCJ4)","x (m)","y (m)","z (m)","vx (m/sec)","vy (m/sec)","vz (m/sec)"
161/2003 00:00:00.00,5851806.472,3742264.707,-85325.570,-3555.922091,5634.811129,3615.150356
161/2003 00:01:00.00,5626064.214,4072094.425,131612.351,-3966.101887,5355.552625,3613.491305
161/2003 00:02:00.00,5376211.062,4384472.905,347984.658,-4359.325961,5053.312847,3596.303120
161/2003 00:03:00.00,5103315.587,4678059.642,562861.404,-4733.899682,4729.387481,3563.656987
161/2003 00:04:00.00,4808545.666,4951594.735,775318.929,-5088.208714,4385.166549,3515.691262
161/2003 00:05:00.00,4493163.450,5203904.362,984443.872,-5420.726193,4022.128298,3452.610885
161/2003 00:06:00.00,4158519.907,5433905.883,1189337.146,-5730.019485,3641.832672,3374.686480
161/2003 00:07:00.00,3806048.977,5640612.534,1389117.839,-6014.756513,3245.914410,3282.253150
161/2003 00:08:00.00,3437261.365,5823137.703,1582927.044,-6273.711599,2836.075810,3175.708973
161/2003 00:09:00.00,3053737.986,5980698.755,1769931.580,-6505.770813,2414.079187,3055.513213
```

File name format:
GyyyydddEPVPRE.vv

Planned Observatory Timeline

This section describes the format of the Planned Observatory file. This file contains the schedule produced by Science Input Processor. It is input to the Timeline Monitor and the Attitude-dependent TDRSS Scheduling software.

Filename Convention

POT_yyyydddhhmm_yyyydddhhmm_vv.txt

yyyydddhhmm is a time stamp where yyyy is a 4-digit year, ddd is the 3-day day of year and hh is hours and mm is minutes. The time stamps indicate the beginning and end times of the scheduling window for which the POT was generated. All events in the timeline occur within this range.

vv is a version number, initially 00. It is used to distinguish different POT files created for the same time range. The highest number is the most recent version.

Example: POT_20030071200_20030101200_00.txt

File Format

A POT record is composed of columns (fields) defined below. TBD

Column	Description	Format
1	Time Of Event	YYYY-DOY-HH:MM:SS
2	Event Type	Text: Point, Survey
3	Begin/End of Event	Text: Begin or End
4	Target Name/Survey mode	Text: Target name or Mode
7	Observation Number	Unsigned Integer (32bits)
8	RA	Floating point (64bits)
9	DEC	Floating point (64 bits)
10	Roll	Floating point (32 bits)
16	Comment	Text

Real-Time Command Log

This section describes the format of the Real-time Command Log file output from the Event Delogger. The Real-Time Command Log is a subset of an ITOS Event Log file. This report only contains lines that contain command type events that are specified by the search criteria, which can be modified in the Event Delogger configuration file.

Filename Convention

The filename is the ITOS event log filename (see ITOS Event Log Definition) prefixed by the letters 'CMD'. For example:

CMDGLASTEVT_02-262-1557.TMP

File Format

The file begins with header lines that describe how the file was created. They include: report title, full pathname of report, time created, ITOS event file name, type (TDRSS or offline) and the search criteria used to select the events. Following the header is either the ITOS events extracted (for format see the ITOS Event Log Definition) or a text message indicating that no events matched the search criteria.

Sample Real-Time Command Log File

```
Spacecraft Command Summary
File: /home/csmith/moc/my_filesvr/ITOS/output/reports/CmdArchive/malindi/CMDGLASTEVT_02-262-1557.TMP
Created: 2003-009-14:33:59
Event file: /home/csmith/test/ed/GLASTEVT_02-262-1557.TMP  Type: malindi
Search criteria: dsp_evtlog:|CMD_VERIFY|CMD_EVENT: ALERT
________________________________________________________________________________________

00 02-262-15:57:59 NULL_EVENT: dsp_evtlog: opened (append) log file /usr/tcw.GLAST/logs/GLASTEVT_02-262-1557.TMP
09 02-262-16:04:43 CMD_VERIFY: fop 1: Verified frame sequence number 0, command /ILIMITON
09 02-262-16:04:44 CMD_VERIFY: fop 1: Verified frame sequence number 1, command /SCNOOP
09 02-262-16:04:44 CMD_VERIFY: fop 1: Verified frame sequence number 2, command /SCATSSTOP
09 02-262-16:04:44 CMD_VERIFY: fop 1: Verified frame sequence number 3, command /SCRTSENABLE RTS00
09 02-262-16:04:44 CMD_VERIFY: fop 1: Verified frame sequence number 4, command /FOSETPPTMODE DISABLE
09 02-262-16:04:45 CMD_VERIFY: fop 1: Verified frame sequence number 5, command /FOSLEWINFOREPLY OBSNUMBER = 0xffff, SLEWREPLY = 5, RA = 2.2, -
20.01,;;
09 02-262-16:55:28 CMD_VERIFY: fop 1: Verified frame sequence number 156, command /FOTOOREQUEST OBSNUMBER=0x2001, TRIGGERSECS = 2001, TRIGGERS
123, MERIT = -20.01,;;
09 02-262-16:55:30 CMD_VERIFY: fop 1: Verified frame sequence number 157, command /ILIMITON
09 02-262-16:55:30 CMD_VERIFY: fop 1: Verified frame sequence number 158, command /SCNOOP
09 02-262-16:55:30 CMD_VERIFY: fop 1: Verified frame sequence number 159, command /SCATSSTOP
00 02-262-16:56:47 NULL_EVENT: dsp_evtlog: closing log file /usr/tcw.GLAST/logs/GLASTEVT_02-262-1557.TMP
Event Delog complete.
```

Appendix C: ACRONYM LIST

ACS	Attitude Control System
AMAC	Automation Monitoring and Control
APID	Application Identifier
ASCII	American Standard Code for Information Interchange
AR	Automated Repoint
ATS	Absolute Time Sequence
BAP	Burst Alert Processor
CCB	Configuration Control Board
CCR	Configuration Change Request
CCSDS	Consultative Committee for Space Data Systems
CDU	Command Data Unit
CGI	Common Gateway Interface
CLCW	Command Link Control Words
CLTU	Command Link Transmission Unit
CMD	Command
COP-1	Command Operations Procedure (version 1)
COTS	Commercial-Off-The-Shelf
CPU	Central Processing Unit
CVT	Current Value Table
DAS	Demand Access System
DB	Database
DEC	Declination
DFD	Data Flow Diagram
DSL	Digital Subscriber Line
DSMC	Data Systems Management Center
DTAS	Data Trending and Analysis System
ETR	Eastern Test Range
FARM	Frame Acceptance and Reporting Mechanism
FDF	Flight Dynamics Facility
FIFO	First in, First out
FITS	Flexible Image Transport System
FOP	Frame Operation Procedure
FOT	Flight Operations Team
FSW	Flight Software
FTP	File Transfer Protocol
Gbits	Gigabits
GBM	Gamma-Ray Burst Monitor
GCN	GRB Coordinates Network
GMT	Greenwich Mean Time

GN	Ground Network
GOTS	Government Off-The-Shelf
GPS	Global Positioning System
GRB	Gamma Ray Burst
GSFC	Goddard Space Flight Center
GSSC	GLAST Science Support Center
GUI	Graphical User Interface
HD	Hard Drive
HEASARC	High Energy Astrophysics Science Archive Research Center
ICD	Interface Control Document
IIRV	Improved Inter-Range Vector
IOC	Instrument Operations Center
IONet	IP Operational Network
ITOS	Integrated Test and Operations System
kbps	Thousand bits per second
KSC	Kennedy Space Center
L0	Level 0 (zero)
LAN	Local Area Network
LAT	Large Area Telescope
LoM	Life of Mission
LOS	Loss of Signal
L&EO	Launch & Early Orbit
M_PDU	Multiplexing Protocol Data Unit
MA	Multiple Access
Mbps	Million bits per second
MOC	GLAST Mission Operations Center
MPS	GLAST Mission Planning System
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communications Network
NISN	NASA Integrated Services Network
NORAD	North American Regional Air Defense
NTP	Network Time Protocol
OCR	Operations Control Room
OIG	Orbital Information Group
PB	Playback
PI	Principal Investigator
RA	Right Ascension
RAM	Random Access Memory
RF	Radio Frequency
RS	Reed-Solomon
RT	Real-time
RTS	Relative Time Sequence
SAA	South Atlantic Anomaly

SERS	Spacecraft Emergency Response System
SFTP	Secure File Transfer Protocol
SLAC	Stanford Linear Accelerator Center
SMEX	Small Explorers
SN	Space Network
SOH	State of Health
SOT	Science Operations Team
SSR	Solid State Recorder
STK	Satellite Tool Kit
SV	State Vector
STOL	Spacecraft Test and Operations Language
SWSI	SN Web Services Interface
TBD	To Be Determined
TBS	To Be Supplied
TC	Telecommand
TCP/IP	Transport Control Protocol/Internet Protocol
TDRS	Tracking and Data Relay Satellite
TDRSS	Tracking and Data Relay Satellite System
TLE	Two-Line Elements
TLM	Telemetry
ToO	Target of Opportunity
TUT	TDRSS Unscheduled Time Report
UDP/IP	User Datagram Protocol/Internet Protocol
UK	United Kingdom
USN	Universal Space Network
UTC	Universal Time Coordinate
VC	Virtual Channel
VMOC	Virtual Mission Operations Center
VR	Virtual Recorder
WDISC	WSC TCP/IP Data Interface Service Capability
WSC	White Sands Complex
WWW	World Wide Web